Hybrid Logics

Carlos Areces
carlos.areces@gmail.com

Spring Term 2018
Stanford

What we want to cover

- Introduce the basic hybrid logics $\mathcal{H}(@)$.

What we want to cover

- Introduce the basic hybrid logics $\mathcal{H}(@)$.
- Extend the standard translation to $\mathcal{H}(@)$.

What we want to cover

- Introduce the basic hybrid logics $\mathcal{H}(@)$.
- Extend the standard translation to $\mathcal{H}(@)$.
- Talk about expressive power. Bisimulation.

What we want to cover

- Introduce the basic hybrid logics $\mathcal{H}(@)$.
- Extend the standard translation to $\mathcal{H}(@)$.
- Talk about expressive power. Bisimulation.
- Define a tableax algorithm for satisfiability of formulas in the basic modal logic (BML).

What we want to cover

- Introduce the basic hybrid logics $\mathcal{H}(@)$.
- Extend the standard translation to $\mathcal{H}(@)$.
- Talk about expressive power. Bisimulation.
- Define a tableax algorithm for satisfiability of formulas in the basic modal logic (BML).
  - How do we check whether a formula has a model?

What we want to cover

- Introduce the basic hybrid logics $\mathcal{H}(@)$.
- Extend the standard translation to $\mathcal{H}(@)$.
- Talk about expressive power. Bisimulation.
- Define a tableax algorithm for satisfiability of formulas in the basic modal logic (BML).
  - How do we check whether a formula has a model?
  - What can we learn from tableaux?

What we want to cover

- Introduce the basic hybrid logics $\mathcal{H}(@)$.
- Extend the standard translation to $\mathcal{H}(@)$.
- Talk about expressive power. Bisimulation.
- Define a tableax algorithm for satisfiability of formulas in the basic modal logic (BML).
  - How do we check whether a formula has a model?
  - What can we learn from tableaux?
- Transform the tableau for BML into one for $\mathcal{H}(@)$.

What we want to cover

- Introduce the basic hybrid logics $\mathcal{H}(@)$.
- Extend the standard translation to $\mathcal{H}(@)$.
- Talk about expressive power. Bisimulation.
- Define a tableax algorithm for satisfiability of formulas in the basic modal logic (BML).
  - How do we check whether a formula has a model?
  - What can we learn from tableaux?
- Transform the tableau for BML into one for $\mathcal{H}(@)$.
- Discuss more expressive hybrid logics.

Relevant Bibliography

- Relevant section from Modal Logic for Open Mind:
  - Section 7.4: Extended Modal Languages

Relevant Bibliography

- Relevant section from Modal Logic for Open Mind:
  - Section 7.4: Extended Modal Languages
- Blackburn, P., de Rijke, M. and Venema, Y. Chapter 7, Section 3 of "Modal Logic", Cambridge Tracts in Theoretical Computer Science, 53, Cambridge University Press, 2001.

Relevant Bibliography

- Relevant section from Modal Logic for Open Mind:
  - Section 7.4: Extended Modal Languages
- Blackburn, P., de Rijke, M. and Venema, Y. Chapter 7, Section 3 of "Modal Logic", Cambridge Tracts in Theoretical Computer Science, 53, Cambridge University Press, 2001.
- Blackburn, P. Representation, Reasoning, and Relational Structures: a Hybrid Logic Manifesto. Logic Journal of the IGPL, 8(3), 339–625, 2000.

Relevant Bibliography

- Relevant section from Modal Logic for Open Mind:
  - Section 7.4: Extended Modal Languages
- Blackburn, P., de Rijke, M. and Venema, Y. Chapter 7, Section 3 of "Modal Logic", Cambridge Tracts in Theoretical Computer Science, 53, Cambridge University Press, 2001.
- Blackburn, P. Representation, Reasoning, and Relational Structures: a Hybrid Logic Manifesto. Logic Journal of the IGPL, 8(3), 339–625, 2000.
- Areces, C. and ten Cate, B.. Hybrid Logics. In Blackburn, P., Wolter, F., and van Benthem, J., editors, Handbook of Modal Logics, 821–868, Elsevier, 2006.

Relevant Bibliography

- Relevant section from Modal Logic for Open Mind:
  - Section 7.4: Extended Modal Languages
- Blackburn, P., de Rijke, M. and Venema, Y. Chapter 7, Section 3 of "Modal Logic", Cambridge Tracts in Theoretical Computer Science, 53, Cambridge University Press, 2001.
- Blackburn, P. Representation, Reasoning, and Relational Structures: a Hybrid Logic Manifesto. Logic Journal of the IGPL, 8(3), 339–625, 2000.
- Areces, C. and ten Cate, B.. Hybrid Logics. In Blackburn, P., Wolter, F., and van Benthem, J., editors, Handbook of Modal Logics, 821–868, Elsevier, 2006.
- Article on "Hybrid Logics" at the Stanford Encyclopedia of Philosophy, `http://plato.stanford.edu/entries/logic-hybrid`

What we want to cover

- Introduce the basic hybrid logics $\mathcal{H}(@)$.
- Extend the standard translation to $\mathcal{H}(@)$.
- Talk about expressive power. Bisimulation.
- Define a tableax algorithm for satisfiability of formulas in the basic modal logic (BML).
  - How do we check whether a formula has a model?
  - What can we learn from tableaux?
- Transform the tableau for BML into one for $\mathcal{H}(@)$.
- Discuss more expressive hybrid logics.

The Basic Modal Logic

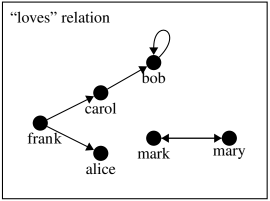
Syntax: Propositional Logic + modalities

Semantics: Interpreted in terms of relational structures (Graphs)

The Basic Modal Logic

Syntax: Propositional Logic + modalities

Semantics: Interpreted in terms of relational structures (Graphs)

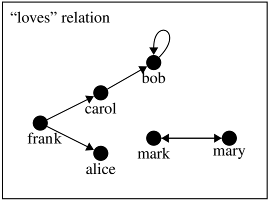


Query: "Does somebody loves a loner?"
(is it true somewhere in the graph?)

The Basic Modal Logic

Syntax: Propositional Logic + modalities

Semantics: Interpreted in terms of relational structures (Graphs)



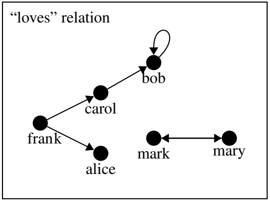
Query: "Does somebody loves a loner?"
(is it true somewhere in the graph?)

The Modal Way: $\varphi := \langle loves \rangle \neg \langle loves \rangle \top$

The Basic Modal Logic

Syntax: Propositional Logic + modalities

Semantics: Interpreted in terms of relational structures (Graphs)



Query: "Does somebody loves a loner?"
(is it true somewhere in the graph?)

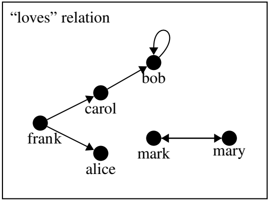
The Modal Way: $\varphi := \langle loves \rangle \neg \langle loves \rangle \top$

- MLs can be (usually) thought of as fragments of FO

The Basic Modal Logic

Syntax: Propositional Logic + modalities

Semantics: Interpreted in terms of relational structures (Graphs)



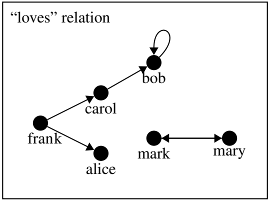
Query: "Does somebody loves a loner?"
(is it true somewhere in the graph?)

The Modal Way: $\varphi := \langle loves \rangle \neg \langle loves \rangle \top$

- MLs can be (usually) thought of as fragments of FO
- Modal logics are (usually) decidable
  - SAT for the basic modal logic is PSpace-complete

The Limits of Modal Expressivity

Some properties can't be expressed in the basic modal language…



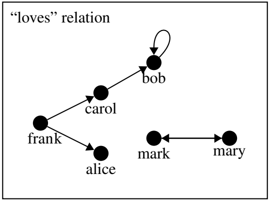
Query: “Does Frank love Alice?”

Query: “Is there somebody who loves himself/herself?”

Query: “Are there two people who loves each other?”

The Limits of Modal Expressivity

Some properties can’t be expressed in the basic modal language…



Query: “Does Frank love Alice?”

Query: “Is there somebody who loves himself/herself?”

Query: “Are there two people who loves each other?”

- What do we need?
  - constants
  - identity

The Limits of Modal Expressivity

- There is an asymmetry at the heart of modal logic:
  - Although states are crucial to modal semantics, nothing in modal syntax can talk about them.
  - Modal logics has no mechanism for referring to or reasoning about the individual states in the structure.

The Limits of Modal Expressivity

- There is an asymmetry at the heart of modal logic:
  - Although states are crucial to modal semantics, nothing in modal syntax can talk about them.
  - Modal logics has no mechanism for referring to or reasoning about the individual states in the structure.
- This leads to two kinds of problem:
  - A representation problem: for some applications modal logic is not adequate as a representation formalism, and
  - A reasoning problem: modal reasoning systems are difficult to devise.

The Limits of Modal Expressivity

- There is an asymmetry at the heart of modal logic:
  - Although states are crucial to modal semantics, nothing in modal syntax can talk about them.
  - Modal logics has no mechanism for referring to or reasoning about the individual states in the structure.
- This leads to two kinds of problem:
  - A representation problem: for some applications modal logic is not adequate as a representation formalism, and
  - A reasoning problem: modal reasoning systems are difficult to devise.
- These limitations motivated the work on Hybrid Logics.

The Basic Recipe: $\mathcal{H}(@)$

basic modal logic

The Basic Recipe: $\mathcal{H}(@)$

basic modal logic

+ nominals $\rightarrow$ a new atomic sort

The Basic Recipe: $\mathcal{H}(@)$

basic modal logic

+	nominals	$\rightarrow$	a new atomic sort
+	@	$\rightarrow$	the ‘at’ operator

The Basic Recipe: $\mathcal{H}(@)$

	basic modal logic		
+	nominals	$\to$	a new atomic sort
+	@	$\to$	the 'at' operator
	$\mathcal{H}(@)$	$\to$	the basic hybrid logic

The Basic Recipe: $\mathcal{H}(@)$

basic modal logic			
+	nominals	$\rightarrow$	a new atomic sort
+	@	$\rightarrow$	the 'at' operator
	$\mathcal{H}(@)$	$\rightarrow$	the basic hybrid logic

- Nominals $i, j, k, \ldots$ denote elements (points) in the model
- $@_i\varphi$ is true iff φ is true at the point denoted by i.

The Basic Recipe: $\mathcal{H}(@)$

basic modal logic			
+ nominals		$\rightarrow$	a new atomic sort
+ @		$\rightarrow$	the 'at' operator
$\mathcal{H}(@)$		$\rightarrow$	the basic hybrid logic

- Nominals $i, j, k, \ldots$ denote elements (points) in the model
- $@_i\varphi$ is true iff φ is true at the point denoted by i.

  For example $@_i p \wedge @_i q \rightarrow @_i(p \wedge q)$ is valid
  while $\Diamond p \wedge \Diamond q \rightarrow \Diamond(p \wedge q)$ is not.

The Basic Recipe: $\mathcal{H}(@)$

basic modal logic			
+ nominals	$\rightarrow$	a new atomic sort
+ @	$\rightarrow$	the 'at' operator
$\mathcal{H}(@)$	$\rightarrow$	the basic hybrid logic

- Nominals $i, j, k, \ldots$ denote elements (points) in the model
- $@_i\varphi$ is true iff φ is true at the point denoted by i.

  For example $@_ip \wedge @_iq \rightarrow @_i(p \wedge q)$ is valid
  while $\Diamond p \wedge \Diamond q \rightarrow \Diamond(p \wedge q)$ is not.

  In particular $@_ij$ says that i and j denote the same point in the model (i.e., $i = j$).

The Hybrid Logic $\mathcal{H}(@)$

Syntax:

$$\text{FORM} := p \mid i \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \Diamond\varphi \mid @_i\varphi, i \in \text{NOM}$$

The Hybrid Logic $\mathcal{H}(@)$

Syntax:

$$\text{FORM} := p \mid i \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \Diamond\varphi \mid @_i\varphi, i \in \text{NOM}$$

Semantics: Use Kripke models $\langle W, R, V\rangle$, add an assignment function $g : \text{NOM} \to W$. We call $\langle W, R, V\rangle, g, w$ a hybrid pointed model.

Define
$$\mathcal{M}, g, w \models i \quad \text{iff} \quad w = g(i) \text{ for } i \in \text{NOM}$$

The Hybrid Logic $\mathcal{H}(@)$

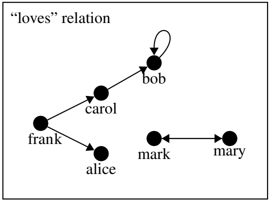
Syntax:

$$\mathsf{FORM} := p \mid i \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \Diamond\varphi \mid @_i\varphi, i \in \mathsf{NOM}$$

Semantics: Use Kripke models $\langle W, R, V\rangle$, add an assignment function $g : \mathsf{NOM} \to W$. We call $\langle W, R, V\rangle, g, w$ a hybrid pointed model.

Define
$$\begin{array}{rcl} \mathcal{M}, g, w \models i & \text{iff} & w = g(i) \text{ for } i \in \mathsf{NOM} \\ \mathcal{M}, g, w \models @_i\varphi & \text{iff} & \mathcal{M}, g, g(i) \models \varphi \end{array}$$

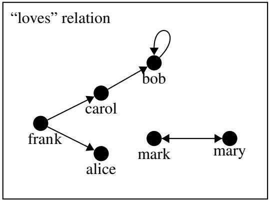
The Expressive Power of $\mathcal{H}(@)$



Query: "Are there two people who loves each other?"

In $\mathcal{H}(@)$: $@_{mark}\langle loves\rangle mary \wedge @_{mary}\langle loves\rangle mark$

The Expressive Power of $\mathcal{H}(@)$

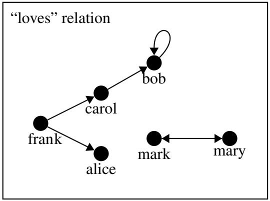


Query: "Are there two people who loves each other?"

In $\mathcal{H}(@)$:

$$@_{mark}\langle loves\rangle mary \wedge$$
$$@_{mary}\langle loves\rangle mark$$
$$\wedge\ @_{mary}\neg mark$$

The Expressive Power of $\mathcal{H}(@)$



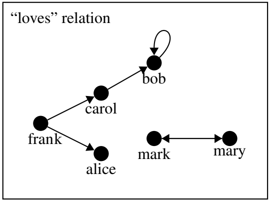
Query: "Are there two people who loves each other?"

In $\mathcal{H}(@)$: $@_{mark}\langle loves\rangle mary \wedge @_{mary}\langle loves\rangle mark \wedge @_{mary}\neg mark$

$\vert\ @_i i$

Congruence. For $i, j, k \in \mathsf{NOM}$:

The Expressive Power of $\mathcal{H}(@)$

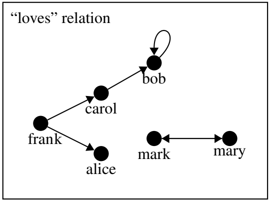


Query: "Are there two people who loves each other?"

In $\mathcal{H}(@)$: $$@_{mark}\langle loves\rangle mary \wedge @_{mary}\langle loves\rangle mark \wedge @_{mary}\neg mark$$

Congruence. For $i, j, k \in \mathsf{NOM}$: $$\begin{array}{|l} @_i i \\ @_i j \rightarrow @_j i \end{array}$$

The Expressive Power of $\mathcal{H}(@)$



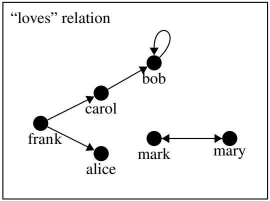
Query: "Are there two people who loves each other?"

In $\mathcal{H}(@)$: $$@_{mark}\langle loves\rangle mary \wedge @_{mary}\langle loves\rangle mark \wedge @_{mary}\neg mark$$

Congruence. For $i, j, k \in \mathsf{NOM}$:

$$@_i i$$
$$@_i j \rightarrow @_j i$$
$$@_i j \wedge @_j k \rightarrow @_i k$$

The Expressive Power of $\mathcal{H}(@)$



Query: "Are there two people who loves each other?"

In $\mathcal{H}(@)$:
$$@_{mark}\langle loves\rangle mary \wedge @_{mary}\langle loves\rangle mark \wedge @_{mary}\neg mark$$

Congruence. For $i, j, k \in \mathsf{NOM}$:

$$@_i i$$
$$@_i j \rightarrow @_j i$$
$$@_i j \wedge @_j k \rightarrow @_i k$$
$$@_i j \rightarrow (@_i \varphi \leftrightarrow @_j \varphi)$$

Hybrid Logics: More examples

Let $i, j \in \mathsf{NOM}$.

- Consider the formula $i \wedge \Diamond i$. What should be true in a model to satisfy it?

Hybrid Logics: More examples

Let $i, j \in \mathsf{NOM}$.

- Consider the formula $i \wedge \Diamond i$. What should be true in a model to satisfy it?
- Consider the formula $\Diamond(i \wedge p) \wedge \Diamond(i \wedge q) \rightarrow \Diamond(p \wedge q)$. Is it valid?

Hybrid Logics: More examples

Let $i, j \in \mathsf{NOM}$.

- Consider the formula $i \wedge \Diamond i$. What should be true in a model to satisfy it?
- Consider the formula $\Diamond(i \wedge p) \wedge \Diamond(i \wedge q) \to \Diamond(p \wedge q)$. Is it valid? What happens if we change i by a $r \in \mathsf{PROP}$?

Hybrid Logics: More examples

Let $i, j \in \mathsf{NOM}$.

- Consider the formula $i \wedge \Diamond i$. What should be true in a model to satisfy it?
- Consider the formula $\Diamond(i \wedge p) \wedge \Diamond(i \wedge q) \rightarrow \Diamond(p \wedge q)$. Is it valid? What happens if we change i by a $r \in \mathsf{PROP}$?
- Consider the formula $@_i \Diamond j \wedge @_j \varphi \rightarrow @_i \Diamond \varphi$. Valid?

What we want to cover

- Introduce the basic hybrid logics $\mathcal{H}(@)$.
- Extend the standard translation to $\mathcal{H}(@)$.
- Talk about expressive power. Bisimulation.
- Define a tableax algorithm for satisfiability of formulas in the basic modal logic (BML).
  - How do we check whether a formula has a model?
  - What can we learn from tableaux?
- Transform the tableau for BML into one for $\mathcal{H}(@)$.
- Discuss more expressive hybrid logics.

The Standard Translation

You have seen the standard translation for BML

$$
\begin{array}{rcl}
\mathsf{ST}_x(p_i) & = & P_i(x) \\
\mathsf{ST}_x(\neg\varphi) & = & \neg\mathsf{ST}_x(\varphi) \\
\mathsf{ST}_x(\varphi \wedge \psi) & = & \mathsf{ST}_x(\varphi) \wedge \mathsf{ST}_x(\psi) \\
\mathsf{ST}_x(\Diamond\varphi) & = & \exists y.(R(x,y) \wedge \mathsf{ST}_y(\varphi)), \quad y \text{ a new variable}
\end{array}
$$

The Standard Translation

You have seen the standard translation for BML

$$
\begin{array}{rcl}
\mathsf{ST}_x(p_i) & = & P_i(x) \\
\mathsf{ST}_x(\neg\varphi) & = & \neg\mathsf{ST}_x(\varphi) \\
\mathsf{ST}_x(\varphi \wedge \psi) & = & \mathsf{ST}_x(\varphi) \wedge \mathsf{ST}_x(\psi) \\
\mathsf{ST}_x(\Diamond\varphi) & = & \exists y.(R(x,y) \wedge \mathsf{ST}_y(\varphi)), \quad y \text{ a new variable}
\end{array}
$$

How do we extend it for $\mathcal{H}(@)$?

The Standard Translation

You have seen the standard translation for BML

$$
\begin{array}{rcl}
\mathsf{ST}_x(p_i) & = & P_i(x) \\
\mathsf{ST}_x(\neg\varphi) & = & \neg\mathsf{ST}_x(\varphi) \\
\mathsf{ST}_x(\varphi \wedge \psi) & = & \mathsf{ST}_x(\varphi) \wedge \mathsf{ST}_x(\psi) \\
\mathsf{ST}_x(\Diamond\varphi) & = & \exists y.(R(x,y) \wedge \mathsf{ST}_y(\varphi)), \quad y \text{ a new variable}
\end{array}
$$

How do we extend it for $\mathcal{H}(@)$? First we have to decide on the correspondence first-order language (FOL). It seems natural to add *NOM* as constant symbols of FOL.

The Standard Translation

You have seen the standard translation for BML

$$\begin{array}{rcl} \mathsf{ST}_x(p_i) & = & P_i(x) \\ \mathsf{ST}_x(\neg\varphi) & = & \neg\mathsf{ST}_x(\varphi) \\ \mathsf{ST}_x(\varphi \wedge \psi) & = & \mathsf{ST}_x(\varphi) \wedge \mathsf{ST}_x(\psi) \\ \mathsf{ST}_x(\Diamond\varphi) & = & \exists y.(R(x,y) \wedge \mathsf{ST}_y(\varphi)), \quad y \text{ a new variable} \end{array}$$

How do we extend it for $\mathcal{H}(@)$? First we have to decide on the correspondence first-order language (FOL). It seems natural to add *NOM* as constant symbols of FOL.Then we can define:

$$\mathsf{ST}_x(i) \quad =$$

The Standard Translation

You have seen the standard translation for BML

$$
\begin{array}{rcl}
\mathsf{ST}_x(p_i) & = & P_i(x) \\
\mathsf{ST}_x(\neg\varphi) & = & \neg\mathsf{ST}_x(\varphi) \\
\mathsf{ST}_x(\varphi \wedge \psi) & = & \mathsf{ST}_x(\varphi) \wedge \mathsf{ST}_x(\psi) \\
\mathsf{ST}_x(\Diamond\varphi) & = & \exists y.(R(x,y) \wedge \mathsf{ST}_y(\varphi)), \quad y \text{ a new variable}
\end{array}
$$

How do we extend it for $\mathcal{H}(@)$? First we have to decide on the correspondence first-order language (FOL). It seems natural to add *NOM* as constant symbols of FOL.Then we can define:

$$\mathsf{ST}_x(i) \quad = \quad x = i$$

The Standard Translation

You have seen the standard translation for BML

$$
\begin{array}{rcl}
\mathsf{ST}_x(p_i) & = & P_i(x) \\
\mathsf{ST}_x(\neg\varphi) & = & \neg\mathsf{ST}_x(\varphi) \\
\mathsf{ST}_x(\varphi \wedge \psi) & = & \mathsf{ST}_x(\varphi) \wedge \mathsf{ST}_x(\psi) \\
\mathsf{ST}_x(\Diamond\varphi) & = & \exists y.(R(x,y) \wedge \mathsf{ST}_y(\varphi)), \quad y \text{ a new variable}
\end{array}
$$

How do we extend it for $\mathcal{H}(@)$? First we have to decide on the correspondence first-order language (FOL). It seems natural to add *NOM* as constant symbols of FOL.Then we can define:

$$
\begin{array}{rcl}
\mathsf{ST}_x(i) & = & x = i \\
\mathsf{ST}_x(@_i\varphi) & = &
\end{array}
$$

The Standard Translation

You have seen the standard translation for BML

$$
\begin{array}{rcl}
\mathsf{ST}_x(p_i) & = & P_i(x) \\
\mathsf{ST}_x(\neg\varphi) & = & \neg\mathsf{ST}_x(\varphi) \\
\mathsf{ST}_x(\varphi \wedge \psi) & = & \mathsf{ST}_x(\varphi) \wedge \mathsf{ST}_x(\psi) \\
\mathsf{ST}_x(\Diamond\varphi) & = & \exists y.(R(x,y) \wedge \mathsf{ST}_y(\varphi)), \quad y \text{ a new variable}
\end{array}
$$

How do we extend it for $\mathcal{H}(@)$? First we have to decide on the correspondence first-order language (FOL). It seems natural to add *NOM* as constant symbols of FOL.Then we can define:

$$
\begin{array}{rcl}
\mathsf{ST}_x(i) & = & x = i \\
\mathsf{ST}_x(@_i\varphi) & = & \mathsf{ST}_i(\varphi)
\end{array}
$$

The Standard Translation

Examples:

$$\mathsf{ST}_x(@_i j) \quad =$$

The Standard Translation

Examples:

$$\mathsf{ST}_x(@_i j) \quad = \quad \mathsf{ST}_i(j)$$

The Standard Translation

Examples:

$$\mathsf{ST}_x(@_i j) = \mathsf{ST}_i(j)$$
$$i = j$$

The Standard Translation

Examples:

$$
\begin{array}{rcl}
\mathsf{ST}_x(@_i j) & = & \mathsf{ST}_i(j) \\
 & & i = j \\
\mathsf{ST}_x(@_i p) & = &
\end{array}
$$

The Standard Translation

Examples:

$$\begin{array}{rcl} \mathsf{ST}_x(@_i j) & = & \mathsf{ST}_i(j) \\ & & i = j \\ \mathsf{ST}_x(@_i p) & = & \mathsf{ST}_i(p) \end{array}$$

The Standard Translation

Examples:

$$\begin{array}{rcl} \mathsf{ST}_x(@_i j) & = & \mathsf{ST}_i(j) \\ & & i = j \\ \mathsf{ST}_x(@_i p) & = & \mathsf{ST}_i(p) \\ & & P(i) \end{array}$$

The Standard Translation

Examples:

$$
\begin{aligned}
\mathsf{ST}_x(@_i j) &= \mathsf{ST}_i(j) \\
&\ i = j \\
\mathsf{ST}_x(@_i p) &= \mathsf{ST}_i(p) \\
&\ P(i) \\
\mathsf{ST}_x(@_i \Diamond j) &=
\end{aligned}
$$

The Standard Translation

Examples:

$$\begin{array}{rcl} \mathsf{ST}_x(@_i j) & = & \mathsf{ST}_i(j) \\ & & i = j \\ \mathsf{ST}_x(@_i p) & = & \mathsf{ST}_i(p) \\ & & P(i) \\ \mathsf{ST}_x(@_i \Diamond j) & = & \mathsf{ST}_i(\Diamond j) \end{array}$$

The Standard Translation

Examples:

$$
\begin{array}{rcl}
\mathsf{ST}_x(@_i j) & = & \mathsf{ST}_i(j) \\
 & & i = j \\
\mathsf{ST}_x(@_i p) & = & \mathsf{ST}_i(p) \\
 & & P(i) \\
\mathsf{ST}_x(@_i \Diamond j) & = & \mathsf{ST}_i(\Diamond j) \\
 & & \exists y.(R(i,y) \wedge ST_y(j))
\end{array}
$$

The Standard Translation

Examples:

$$
\begin{aligned}
\mathsf{ST}_x(@_i j) &= \mathsf{ST}_i(j) \\
& i = j \\
\mathsf{ST}_x(@_i p) &= \mathsf{ST}_i(p) \\
& P(i) \\
\mathsf{ST}_x(@_i \Diamond j) &= \mathsf{ST}_i(\Diamond j) \\
& \exists y.(R(i,y) \wedge ST_y(j)) \\
& \exists y.(R(i,y) \wedge y = j)
\end{aligned}
$$

The Standard Translation

Examples:

$$
\begin{array}{rcl}
\mathsf{ST}_x(@_i j) & = & \mathsf{ST}_i(j) \\
& & i = j \\
\mathsf{ST}_x(@_i p) & = & \mathsf{ST}_i(p) \\
& & P(i) \\
\mathsf{ST}_x(@_i \Diamond j) & = & \mathsf{ST}_i(\Diamond j) \\
& & \exists y.(R(i,y) \wedge ST_y(j)) \\
& & \exists y.(R(i,y) \wedge y = j) \\
& \equiv & R(i,j)
\end{array}
$$

Given $\mathcal{M} = \langle W, R, V\rangle, g$ a hybrid model, let $\mathcal{M}'$ be the corresponding first-order model where relations symbols are interpreted using R and V, and g is used to interpret constant symbols.

The Standard Translation

Examples:

$$
\begin{array}{rcl}
\mathsf{ST}_x(@_i j) & = & \mathsf{ST}_i(j) \\
 & & i = j \\
\mathsf{ST}_x(@_i p) & = & \mathsf{ST}_i(p) \\
 & & P(i) \\
\mathsf{ST}_x(@_i \Diamond j) & = & \mathsf{ST}_i(\Diamond j) \\
 & & \exists y.(R(i,y) \wedge ST_y(j)) \\
 & & \exists y.(R(i,y) \wedge y = j) \\
 & \equiv & R(i,j)
\end{array}
$$

Given $\mathcal{M} = \langle W, R, V \rangle, g$ a hybrid model, let $\mathcal{M}'$ be the corresponding first-order model where relations symbols are interpreted using R and V, and g is used to interpret constant symbols.

Theorem: $\mathcal{M}, g, w \models \varphi$ iff $\mathcal{M}', f[x \mapsto w] \models \mathsf{ST}_x(\varphi)$.

What we want to cover

- Introduce the basic hybrid logics $\mathcal{H}(@)$.
- Extend the standard translation to $\mathcal{H}(@)$.
- Talk about expressive power. Bisimulation.
- Define a tableax algorithm for satisfiability of formulas in the basic modal logic (BML).
  - How do we check whether a formula has a model?
  - What can we learn from tableaux?
- Transform the tableau for BML into one for $\mathcal{H}(@)$.
- Discuss more expressive hybrid logics.

Back to Expressive Power: Bisimulations

Back to Expressive Power: Bisimulations

The standard tool to measure expressive power of modal logics is bisimulation.

Back to Expressive Power: Bisimulations

The standard tool to measure expressive power of modal logics is bisimulation.

Definition. A bisimulation for BML is a non empty binary relation Z between the domains of two models $\mathcal{M}_1 = \langle W_1, R_1, V_1 \rangle$ and $\mathcal{M}_2 = \langle W_2, R_2, V_2 \rangle$ s.t.

Back to Expressive Power: Bisimulations

The standard tool to measure expressive power of modal logics is bisimulation.

Definition. A bisimulation for BML is a non empty binary relation Z between the domains of two models $\mathcal{M}_1 = \langle W_1, R_1, V_1 \rangle$ and $\mathcal{M}_2 = \langle W_2, R_2, V_2 \rangle$ s.t.

- Harmony: If $w_1 Z w_2$ then $w_1 \in V(p)$ iff $w_2 \in V(p)$.

Back to Expressive Power: Bisimulations

The standard tool to measure expressive power of modal logics is bisimulation.

Definition. A bisimulation for BML is a non empty binary relation Z between the domains of two models $\mathcal{M}_1 = \langle W_1, R_1, V_1 \rangle$ and $\mathcal{M}_2 = \langle W_2, R_2, V_2 \rangle$ s.t.

- Harmony: If $w_1 Z w_2$ then $w_1 \in V(p)$ iff $w_2 \in V(p)$.
- Zig: If $w_1 Z w_2$ and $w_1 R_1 v_1$ then there is v_2 s.t. $w_2 R_2 v_2$ and $v_1 Z v_2$.

Back to Expressive Power: Bisimulations

The standard tool to measure expressive power of modal logics is bisimulation.

Definition. A bisimulation for BML is a non empty binary relation Z between the domains of two models $\mathcal{M}_1 = \langle W_1, R_1, V_1 \rangle$ and $\mathcal{M}_2 = \langle W_2, R_2, V_2 \rangle$ s.t.

- Harmony: If $w_1 Z w_2$ then $w_1 \in V(p)$ iff $w_2 \in V(p)$.
- Zig: If $w_1 Z w_2$ and $w_1 R_1 v_1$ then there is v_2 s.t. $w_2 R_2 v_2$ and $v_1 Z v_2$.
- Zag: If $w_1 Z w_2$ and $w_2 R_2 v_2$ then there is v_1 s.t. $w_1 R_1 v_1$ and $v_1 Z v_2$.

Back to Expressive Power: Bisimulations

The standard tool to measure expressive power of modal logics is bisimulation.

Definition. A bisimulation for BML is a non empty binary relation Z between the domains of two models $\mathcal{M}_1 = \langle W_1, R_1, V_1 \rangle$ and $\mathcal{M}_2 = \langle W_2, R_2, V_2 \rangle$ s.t.

- Harmony: If $w_1 Z w_2$ then $w_1 \in V(p)$ iff $w_2 \in V(p)$.
- Zig: If $w_1 Z w_2$ and $w_1 R_1 v_1$ then there is v_2 s.t. $w_2 R_2 v_2$ and $v_1 Z v_2$.
- Zag: If $w_1 Z w_2$ and $w_2 R_2 v_2$ then there is v_1 s.t. $w_1 R_1 v_1$ and $v_1 Z v_2$.

Theorem: If Z is a bisimulation and $w_1 Z w_2$ then $\mathcal{M}_1, w_1 \models \varphi$ iff $\mathcal{M}_2, w_2 \models \varphi$.

Back to Expressive Power: Bisimulations

The standard tool to measure expressive power of modal logics is bisimulation.

Definition. A bisimulation for BML is a non empty binary relation Z between the domains of two models $\mathcal{M}_1 = \langle W_1, R_1, V_1 \rangle$ and $\mathcal{M}_2 = \langle W_2, R_2, V_2 \rangle$ s.t.

- Harmony: If $w_1 Z w_2$ then $w_1 \in V(p)$ iff $w_2 \in V(p)$.
- Zig: If $w_1 Z w_2$ and $w_1 R_1 v_1$ then there is v_2 s.t. $w_2 R_2 v_2$ and $v_1 Z v_2$.
- Zag: If $w_1 Z w_2$ and $w_2 R_2 v_2$ then there is v_1 s.t. $w_1 R_1 v_1$ and $v_1 Z v_2$.

Theorem: If Z is a bisimulation and $w_1 Z w_2$ then $\mathcal{M}_1, w_1 \models \varphi$ iff $\mathcal{M}_2, w_2 \models \varphi$.

This is not the correct notion of bisimulation for, e.g., $BML + \Diamond^{-}$.

Back to Expressive Power: Bisimulations

The standard tool to measure expressive power of modal logics is bisimulation.

Definition. A bisimulation for BML is a non empty binary relation Z between the domains of two models $\mathcal{M}_1 = \langle W_1, R_1, V_1 \rangle$ and $\mathcal{M}_2 = \langle W_2, R_2, V_2 \rangle$ s.t.

- Harmony: If $w_1 Z w_2$ then $w_1 \in V(p)$ iff $w_2 \in V(p)$.
- Zig: If $w_1 Z w_2$ and $w_1 R_1 v_1$ then there is v_2 s.t. $w_2 R_2 v_2$ and $v_1 Z v_2$.
- Zag: If $w_1 Z w_2$ and $w_2 R_2 v_2$ then there is v_1 s.t. $w_1 R_1 v_1$ and $v_1 Z v_2$.

Theorem: If Z is a bisimulation and $w_1 Z w_2$ then $\mathcal{M}_1, w_1 \models \varphi$ iff $\mathcal{M}_2, w_2 \models \varphi$.

This is not the correct notion of bisimulation for, e.g., $BML + \Diamond^-$. Why?

Back to Expressive Power: Bisimulations

The standard tool to measure expressive power of modal logics is bisimulation.

Definition. A bisimulation for BML is a non empty binary relation Z between the domains of two models $\mathcal{M}_1 = \langle W_1, R_1, V_1 \rangle$ and $\mathcal{M}_2 = \langle W_2, R_2, V_2 \rangle$ s.t.

- Harmony: If $w_1 Z w_2$ then $w_1 \in V(p)$ iff $w_2 \in V(p)$.
- Zig: If $w_1 Z w_2$ and $w_1 R_1 v_1$ then there is v_2 s.t. $w_2 R_2 v_2$ and $v_1 Z v_2$.
- Zag: If $w_1 Z w_2$ and $w_2 R_2 v_2$ then there is v_1 s.t. $w_1 R_1 v_1$ and $v_1 Z v_2$.

Theorem: If Z is a bisimulation and $w_1 Z w_2$ then $\mathcal{M}_1, w_1 \models \varphi$ iff $\mathcal{M}_2, w_2 \models \varphi$.

This is not the correct notion of bisimulation for, e.g., $BML + \Diamond^-$. Why? What is the correct notion of bisimulation for $\mathcal{H}(@)$?

Bisimulations for $\mathcal{H}@$

- Consider BML extended with nominals. Do we need any change to the notion of bisimulation?

Bisimulations for $\mathcal{H}$@

- Consider BML extended with nominals. Do we need any change to the notion of bisimulation? No, the Harmony condition takes care of nominals.

Bisimulations for $\mathcal{H}$@

- Consider BML extended with nominals. Do we need any change to the notion of bisimulation? No, the Harmony condition takes care of nominals.
- Consider BML extended with @. Do we need any change?

Bisimulations for $\mathcal{H}@$

- Consider BML extended with nominals. Do we need any change to the notion of bisimulation? No, the Harmony condition takes care of nominals.
- Consider BML extended with @. Do we need any change? Yes, $@_i$ moves evaluation to the point named by i, that might not be reachable by the Zig/Zag conditions.

Bisimulations for $\mathcal{H}@$

- Consider BML extended with nominals. Do we need any change to the notion of bisimulation? No, the Harmony condition takes care of nominals.
- Consider BML extended with @. Do we need any change? Yes, $@_i$ moves evaluation to the point named by i, that might not be reachable by the Zig/Zag conditions.
- Bisimulation for $\mathcal{H}(@)$: To the conditions for BML-bisimulation add the followng conditions:
  - For all $i \in \mathsf{NOM}$, $g_1(i)\ Z\ g_2(i)$.

Bisimulations for $\mathcal{H}@$

- Consider BML extended with nominals. Do we need any change to the notion of bisimulation? No, the Harmony condition takes care of nominals.
- Consider BML extended with @. Do we need any change? Yes, $@_i$ moves evaluation to the point named by i, that might not be reachable by the Zig/Zag conditions.
- Bisimulation for $\mathcal{H}(@)$: To the conditions for BML-bisimulation add the followng conditions:
  - For all $i \in \mathsf{NOM}$, $g_1(i)\ Z\ g_2(i)$.

T: Let $\mathcal{M}_1, g_1$ and $\mathcal{M}_2, g_2$ be $\mathcal{H}@$-bisimilar models (via Z), then $w_1 Z w_2$ implies $\mathcal{M}_1, g_1, w_1 \models \varphi$ iff $\mathcal{M}_2, g_2, w_2 \models \varphi$.

Bisimulations for $\mathcal{H}$@

- Consider BML extended with nominals. Do we need any change to the notion of bisimulation? No, the Harmony condition takes care of nominals.
- Consider BML extended with @. Do we need any change? Yes, $@_i$ moves evaluation to the point named by i, that might not be reachable by the Zig/Zag conditions.
- Bisimulation for $\mathcal{H}(@)$: To the conditions for BML-bisimulation add the followng conditions:
  - For all $i \in \mathsf{NOM}$, $g_1(i)\ Z\ g_2(i)$.

T: Let $\mathcal{M}_1, g_1$ and $\mathcal{M}_2, g_2$ be $\mathcal{H}$@-bisimilar models (via Z), then $w_1 Z w_2$ implies $\mathcal{M}_1, g_1, w_1 \models \varphi$ iff $\mathcal{M}_2, g_2, w_2 \models \varphi$.

- The converse holds on finite-image models (as with BML).
- A "van Benthem Characterization Theorem" holds using the notion of bisimulation for $\mathcal{H}(@)$.

Something to ponder

- We were able to say that *mark* and *mary* loves each other (how nice!). But can we say (in $\mathcal{H}(@)$) that there are two people who loves each other (i.e., without naming them)?

Something to ponder

- We were able to say that *mark* and *mary* loves each other (how nice!). But can we say (in $\mathcal{H}(@)$) that there are two people who loves each other (i.e., without naming them)?
- Can we say that somebody loves himself? (again, if we don't know his/her name).

Something to ponder

- We were able to say that *mark* and *mary* loves each other (how nice!). But can we say (in $\mathcal{H}(@)$) that there are two people who loves each other (i.e., without naming them)?
- Can we say that somebody loves himself? (again, if we don't know his/her name).
- Think about it for next class.

What we want to cover

- Introduce the basic hybrid logics $\mathcal{H}(@)$.
- Extend the standard translation to $\mathcal{H}(@)$.
- Talk about expressive power. Bisimulation.
- Define a tableax algorithm for satisfiability of formulas in the basic modal logic (BML).
  - How do we check whether a formula has a model?
  - What can we learn from tableaux?
- Transform the tableau for BML into one for $\mathcal{H}(@)$.
- Discuss more expressive hybrid logics.

Enumerating Models

- The proof that the satisfiability problem for PL is decidable is very simple:
  - Suppose that you are given a formula φ and you are looking for a model of φ.

Enumerating Models

- The proof that the satisfiability problem for PL is decidable is very simple:
  - Suppose that you are given a formula φ and you are looking for a model of φ.
  - First note that propositional symbols that do not appear in φ are irrelevant.

Enumerating Models

- The proof that the satisfiability problem for PL is decidable is very simple:
  - Suppose that you are given a formula φ and you are looking for a model of φ.
  - First note that propositional symbols that do not appear in φ are irrelevant.
  - Then note that models for PL have only one point.

Enumerating Models

- The proof that the satisfiability problem for PL is decidable is very simple:
  - Suppose that you are given a formula φ and you are looking for a model of φ.
  - First note that propositional symbols that do not appear in φ are irrelevant.
  - Then note that models for PL have only one point.
  - Hence, we only need to list all possible ways of labelling that single point with propositional symbols in φ.

Enumerating Models

- The proof that the satisfiability problem for PL is decidable is very simple:
  - Suppose that you are given a formula φ and you are looking for a model of φ.
  - First note that propositional symbols that do not appear in φ are irrelevant.
  - Then note that models for PL have only one point.
  - Hence, we only need to list all possible ways of labelling that single point with propositional symbols in φ.
- What about the basic modal language?

The Tableaux Method for Relational Structures

The Tableaux Method for Relational Structures

- We want to devise a tableau method for the language we introduced to talk about complex relational structures.

The Tableaux Method for Relational Structures

- We want to devise a tableau method for the language we introduced to talk about complex relational structures.
- Let's review the tableaux method that we introduced for propositional logic:

$$\frac{(\varphi \wedge \psi)}{\begin{array}{c}\varphi \\ \psi\end{array}} \; (\wedge)$$

$$\frac{\neg(\varphi \wedge \psi)}{\neg\varphi \qquad\qquad \neg\psi} \; (\neg\wedge)$$

$$\frac{\neg\neg\varphi}{\varphi} \; (\neg\neg)$$

The Tableaux Method for Relational Structures

- We want to devise a tableau method for the language we introduced to talk about complex relational structures.
- Let's review the tableaux method that we introduced for propositional logic:

$$\frac{(\varphi \land \psi)}{\begin{array}{c}\varphi\\ \psi\end{array}}\ (\land)$$

$$\frac{\neg(\varphi \land \psi)}{\neg\varphi \qquad\qquad \neg\psi}\ (\neg\land)$$

$$\frac{\neg\neg\varphi}{\varphi}\ (\neg\neg)$$

- Pretty neat: 3 rules for an NP-complete problem!

The Tableaux Method for Relational Structures

- We want to devise a tableau method for the language we introduced to talk about complex relational structures.
- Let's review the tableaux method that we introduced for propositional logic:

$$\frac{(\varphi \wedge \psi)}{\begin{array}{c}\varphi \\ \psi\end{array}}\ (\wedge)$$

$$\frac{\neg(\varphi \wedge \psi)}{\neg\varphi \qquad\qquad \neg\psi}\ (\neg\wedge)$$

$$\frac{\neg\neg\varphi}{\varphi}\ (\neg\neg)$$

- Pretty neat: 3 rules for an NP-complete problem!
- But now we want to deal with more than a single point.

The Tableaux Method for Relational Structures

- We want to devise a tableau method for the language we introduced to talk about complex relational structures.
- Let's review the tableaux method that we introduced for propositional logic:

$$\frac{s{:}(\varphi \wedge \psi)}{\begin{array}{c} s{:}\varphi \\ s{:}\psi \end{array}} \ (\wedge)$$

$$\frac{s{:}\neg(\varphi \wedge \psi)}{s{:}\neg\varphi \qquad s{:}\neg\psi} \ (\neg\wedge)$$

$$\frac{s{:}\neg\neg\varphi}{s{:}\varphi} \ (\neg\neg)$$

- Pretty neat: 3 rules for an NP-complete problem!
- But now we want to deal with more than a single point.
- The solution is: labels!

The Tableaux Method for Relational Structures

- We want to devise a tableau method for the language we introduced to talk about complex relational structures.
- Let's review the tableaux method that we introduced for propositional logic:

$$\frac{s{:}(\varphi \wedge \psi)}{\begin{array}{c} s{:}\varphi \\ s{:}\psi \end{array}}\ (\wedge)$$

$$\frac{s{:}\neg(\varphi \wedge \psi)}{s{:}\neg\varphi \qquad s{:}\neg\psi}\ (\neg\wedge)$$

$$\frac{s{:}\neg\neg\varphi}{s{:}\varphi}\ (\neg\neg)$$

- Pretty neat: 3 rules for an NP-complete problem!
- But now we want to deal with more than a single point.
- The solution is: labels!
- They will help us keep track of what is going on in each point in our model.

Now Lines!

Now Lines!

- We have dealt in the previous slide with multiple points. What about lines?

Now Lines!

- We have dealt in the previous slide with multiple points. What about lines?
- Remember that the operator we introduced to talk about lines in our language was $\Diamond\varphi$ and we said that

$$\mathcal{M}, w \models \Diamond\varphi \text{ iff there is } w' \text{ s.t. } wRw' \text{ and } \mathcal{M}, w' \models \varphi.$$

Now Lines!

- We have dealt in the previous slide with multiple points. What about lines?
- Remember that the operator we introduced to talk about lines in our language was $\Diamond\varphi$ and we said that

$$\mathcal{M}, w \models \Diamond\varphi \text{ iff there is } w' \text{ s.t. } wRw' \text{ and } \mathcal{M}, w' \models \varphi.$$

- Start with the labelled formula $s{:}\Diamond\varphi$. ⟶

$$\frac{s{:}\Diamond\varphi}{\begin{array}{c} sRt \\ t{:}\varphi \end{array}} \; (\Diamond)$$

for t a new label

Now Lines!

- We have dealt in the previous slide with multiple points. What about lines?
- Remember that the operator we introduced to talk about lines in our language was $\Diamond\varphi$ and we said that

$$\mathcal{M}, w \models \Diamond\varphi \text{ iff there is } w' \text{ s.t. } wRw' \text{ and } \mathcal{M}, w' \models \varphi.$$

- Start with the labelled formula $s{:}\Diamond\varphi$.
  If this formula is satisfiable, it is because there is an R-sucessor t where φ holds.

$$\frac{s{:}\Diamond\varphi}{\begin{array}{c} sRt \\ t{:}\varphi \end{array}} \; (\Diamond)$$

for t a new label

Now Lines!

- We have dealt in the previous slide with multiple points. What about lines?
- Remember that the operator we introduced to talk about lines in our language was $\Diamond\varphi$ and we said that

$$\mathcal{M}, w \models \Diamond\varphi \text{ iff there is } w' \text{ s.t. } wRw' \text{ and } \mathcal{M}, w' \models \varphi.$$

- Start with the labelled formula $s{:}\Diamond\varphi$. If this formula is satisfiable, it is because there is an R-sucessor t where φ holds.

$$\frac{s{:}\Diamond\varphi}{\begin{array}{c} sRt \\ t{:}\varphi \end{array}} \ (\Diamond)$$

for t a new label

- Start with the labelled formula $s{:}\neg\Diamond\varphi$. ⟶

$$\frac{\begin{array}{c} s{:}\neg\Diamond\varphi \\ sRt \end{array}}{t{:}\neg\varphi} \ (\neg\Diamond)$$

Now Lines!

- We have dealt in the previous slide with multiple points. What about lines?
- Remember that the operator we introduced to talk about lines in our language was $\Diamond\varphi$ and we said that

$$\mathcal{M}, w \models \Diamond\varphi \text{ iff there is } w' \text{ s.t. } wRw' \text{ and } \mathcal{M}, w' \models \varphi.$$

- Start with the labelled formula $s{:}\Diamond\varphi$.
  If this formula is satisfiable, it is because there is an R-sucessor t where φ holds.

$$\frac{s{:}\Diamond\varphi}{\begin{array}{c} sRt \\ t{:}\varphi \end{array}}\ (\Diamond)$$

for t a new label

- Start with the labelled formula $s{:}\neg\Diamond\varphi$.
  If there is an R-successor t, then φ should not hold at t.

$$\frac{\begin{array}{c} s{:}\neg\Diamond\varphi \\ sRt \end{array}}{t{:}\neg\varphi}\ (\neg\Diamond)$$

Tableaux for Modal Logics

- Clash: $s{:}p$ and $s{:}\neg p$ in a branch.

Tableaux for Modal Logics

- Clash: $s{:}p$ and $s{:}\neg p$ in a branch.
- To check satisfiability of φ, start the tableaux with the labelled formula $s{:}\varphi$.

Tableaux for Modal Logics

- Clash: $s{:}p$ and $s{:}\neg p$ in a branch.
- To check satisfiability of φ, start the tableaux with the labelled formula $s{:}\varphi$.
- The previous 5 rules provide a sound and complete calculus for the basic modal logic
- It is actually terminating, and by imposing some restrictions on applications it can run in PSPACE, so it is optimal.

Tableaux for Modal Logics

- Clash: $s{:}p$ and $s{:}\neg p$ in a branch.
- To check satisfiability of φ, start the tableaux with the labelled formula $s{:}\varphi$.
- The previous 5 rules provide a sound and complete calculus for the basic modal logic
- It is actually terminating, and by imposing some restrictions on applications it can run in PSPACE, so it is optimal.
- To think: what is the status of labeled formulas $s{:}\varphi$ and accessibility statements sRt?

The Complete Cast, plus an Example

The Complete Cast, plus an Example

$$\frac{s{:}(\varphi \wedge \psi)}{\begin{array}{c} s{:}\varphi \\ s{:}\psi \end{array}}$$

$$\frac{s{:}\Diamond\varphi}{\begin{array}{c} sRt \\ t{:}\varphi \end{array}}$$

for t a new label

$$\frac{s{:}\neg(\varphi \wedge \psi)}{s{:}\neg\varphi \qquad s{:}\neg\psi}$$

$$\frac{\begin{array}{c} s{:}\neg\Diamond\varphi \\ sRt \end{array}}{t{:}\neg\varphi}$$

$$\frac{s{:}\neg\neg\varphi}{s{:}\varphi}$$

The Complete Cast, plus an Example

$$\frac{s{:}(\varphi \wedge \psi)}{\begin{array}{c} s{:}\varphi \\ s{:}\psi \end{array}}$$

$$\frac{s{:}\neg(\varphi \wedge \psi)}{s{:}\neg\varphi \qquad s{:}\neg\psi}$$

$$\frac{s{:}\neg\neg\varphi}{s{:}\varphi}$$

$$\frac{s{:}\Diamond\varphi}{\begin{array}{c} sRt \\ t{:}\varphi \end{array}}$$

for t a new label

$$\frac{\begin{array}{c} s{:}\neg\Diamond\varphi \\ sRt \end{array}}{t{:}\neg\varphi}$$

$$s{:}(\Diamond p \wedge (\neg\Diamond\neg q \wedge \neg\Diamond(p \wedge q)))$$

The Complete Cast, plus an Example

$$\frac{s{:}(\varphi \wedge \psi)}{\begin{array}{c} s{:}\varphi \\ s{:}\psi \end{array}}$$

$$\frac{s{:}\neg(\varphi \wedge \psi)}{s{:}\neg\varphi \qquad s{:}\neg\psi}$$

$$\frac{s{:}\neg\neg\varphi}{s{:}\varphi}$$

$$\frac{s{:}\Diamond\varphi}{\begin{array}{c} sRt \\ t{:}\varphi \end{array}}$$

for t a new label

$$\frac{\begin{array}{c} s{:}\neg\Diamond\varphi \\ sRt \end{array}}{t{:}\neg\varphi}$$

$$s{:}(\Diamond p \wedge (\neg\Diamond\neg q \wedge \neg\Diamond(p \wedge q)))$$

The Complete Cast, plus an Example

$$\frac{s{:}(\varphi \wedge \psi)}{\begin{array}{c} s{:}\varphi \\ s{:}\psi \end{array}}$$

$$\frac{s{:}\neg(\varphi \wedge \psi)}{s{:}\neg\varphi \qquad s{:}\neg\psi}$$

$$\frac{s{:}\neg\neg\varphi}{s{:}\varphi}$$

$$\frac{s{:}\Diamond\varphi}{\begin{array}{c} sRt \\ t{:}\varphi \end{array}}$$

for t a new label

$$\frac{\begin{array}{c} s{:}\neg\Diamond\varphi \\ sRt \end{array}}{t{:}\neg\varphi}$$

$$s{:}(\Diamond p \wedge (\neg\Diamond\neg q \wedge \neg\Diamond(p \wedge q)))$$
$$s{:}\Diamond p$$
$$s{:}\neg\Diamond\neg q \wedge \neg\Diamond(p \wedge q)$$

The Complete Cast, plus an Example

$$\frac{s{:}(\varphi \land \psi)}{\begin{array}{c} s{:}\varphi \\ s{:}\psi \end{array}}$$

$$\frac{s{:}\Diamond\varphi}{\begin{array}{c} sRt \\ t{:}\varphi \end{array}}$$

for t a new label

$$\frac{s{:}\neg(\varphi \land \psi)}{s{:}\neg\varphi \qquad s{:}\neg\psi}$$

$$\frac{\begin{array}{c} s{:}\neg\Diamond\varphi \\ sRt \end{array}}{t{:}\neg\varphi}$$

$$\frac{s{:}\neg\neg\varphi}{s{:}\varphi}$$

$$s{:}(\Diamond p \land (\neg\Diamond\neg q \land \neg\Diamond(p \land q)))$$
$$s{:}\Diamond p$$
$$s{:}\neg\Diamond\neg q \land \neg\Diamond(p \land q)$$

The Complete Cast, plus an Example

$$\frac{s{:}(\varphi \wedge \psi)}{\begin{array}{c} s{:}\varphi \\ s{:}\psi \end{array}}$$

$$\frac{s{:}\neg(\varphi \wedge \psi)}{s{:}\neg\varphi \qquad s{:}\neg\psi}$$

$$\frac{s{:}\neg\neg\varphi}{s{:}\varphi}$$

$$\frac{s{:}\Diamond\varphi}{\begin{array}{c} sRt \\ t{:}\varphi \end{array}}$$

for t a new label

$$\frac{\begin{array}{c} s{:}\neg\Diamond\varphi \\ sRt \end{array}}{t{:}\neg\varphi}$$

$$\begin{array}{c} s{:}(\Diamond p \wedge (\neg\Diamond\neg q \wedge \neg\Diamond(p \wedge q))) \\ s{:}\Diamond p \\ s{:}\neg\Diamond\neg q \wedge \neg\Diamond(p \wedge q) \\ s{:}\neg\Diamond\neg q \\ s{:}\neg\Diamond(p \wedge q) \end{array}$$

The Complete Cast, plus an Example

$$\frac{s{:}(\varphi \wedge \psi)}{\begin{array}{c} s{:}\varphi \\ s{:}\psi \end{array}}$$

$$\frac{s{:}\neg(\varphi \wedge \psi)}{s{:}\neg\varphi \qquad s{:}\neg\psi}$$

$$\frac{s{:}\neg\neg\varphi}{s{:}\varphi}$$

$$\frac{s{:}\Diamond\varphi}{\begin{array}{c} sRt \\ t{:}\varphi \end{array}}$$

for t a new label

$$\frac{\begin{array}{c} s{:}\neg\Diamond\varphi \\ sRt \end{array}}{t{:}\neg\varphi}$$

$$s{:}(\Diamond p \wedge (\neg\Diamond\neg q \wedge \neg\Diamond(p \wedge q)))$$
$$s{:}\Diamond p$$
$$s{:}\neg\Diamond\neg q \wedge \neg\Diamond(p \wedge q)$$
$$s{:}\neg\Diamond\neg q$$
$$s{:}\neg\Diamond(p \wedge q)$$

The Complete Cast, plus an Example

$$\frac{s{:}(\varphi \wedge \psi)}{\begin{array}{c} s{:}\varphi \\ s{:}\psi \end{array}}$$

$$\frac{s{:}\Diamond\varphi}{\begin{array}{c} sRt \\ t{:}\varphi \end{array}}$$

for t a new label

$$\frac{s{:}\neg(\varphi \wedge \psi)}{s{:}\neg\varphi \qquad s{:}\neg\psi}$$

$$\frac{\begin{array}{c} s{:}\neg\Diamond\varphi \\ sRt \end{array}}{t{:}\neg\varphi}$$

$$\frac{s{:}\neg\neg\varphi}{s{:}\varphi}$$

$$\begin{array}{c} s{:}(\Diamond p \wedge (\neg\Diamond\neg q \wedge \neg\Diamond(p \wedge q))) \\ s{:}\Diamond p \\ s{:}\neg\Diamond\neg q \wedge \neg\Diamond(p \wedge q) \\ s{:}\neg\Diamond\neg q \\ s{:}\neg\Diamond(p \wedge q) \\ sRt \\ t{:}p \end{array}$$

The Complete Cast, plus an Example

$$\frac{s{:}(\varphi \wedge \psi)}{\begin{array}{c} s{:}\varphi \\ s{:}\psi \end{array}}$$

$$\frac{s{:}\neg(\varphi \wedge \psi)}{s{:}\neg\varphi \qquad s{:}\neg\psi}$$

$$\frac{s{:}\neg\neg\varphi}{s{:}\varphi}$$

$$\frac{s{:}\Diamond\varphi}{\begin{array}{c} sRt \\ t{:}\varphi \end{array}}$$

for t a new label

$$\frac{\begin{array}{c} s{:}\neg\Diamond\varphi \\ sRt \end{array}}{t{:}\neg\varphi}$$

$$\begin{array}{c} s{:}(\Diamond p \wedge (\neg\Diamond\neg q \wedge \neg\Diamond(p \wedge q))) \\ s{:}\Diamond p \\ s{:}\neg\Diamond\neg q \wedge \neg\Diamond(p \wedge q) \\ s{:}\neg\Diamond\neg q \\ s{:}\neg\Diamond(p \wedge q) \\ sRt \\ t{:}p \end{array}$$

The Complete Cast, plus an Example

$$\frac{s{:}(\varphi \land \psi)}{\begin{array}{c} s{:}\varphi \\ s{:}\psi \end{array}}$$

$$\frac{s{:}\neg(\varphi \land \psi)}{s{:}\neg\varphi \qquad s{:}\neg\psi}$$

$$\frac{s{:}\neg\neg\varphi}{s{:}\varphi}$$

$$\frac{s{:}\Diamond\varphi}{\begin{array}{c} sRt \\ t{:}\varphi \end{array}}$$

for t a new label

$$\frac{\begin{array}{c} s{:}\neg\Diamond\varphi \\ sRt \end{array}}{t{:}\neg\varphi}$$

$$s{:}(\Diamond p \land (\neg\Diamond\neg q \land \neg\Diamond(p \land q)))$$
$$s{:}\Diamond p$$
$$s{:}\neg\Diamond\neg q \land \neg\Diamond(p \land q)$$
$$s{:}\neg\Diamond\neg q$$
$$s{:}\neg\Diamond(p \land q)$$
$$sRt$$
$$t{:}p$$
$$t{:}\neg\neg q$$

The Complete Cast, plus an Example

$$\frac{s{:}(\varphi \wedge \psi)}{\begin{array}{c} s{:}\varphi \\ s{:}\psi \end{array}}$$

$$\frac{s{:}\neg(\varphi \wedge \psi)}{s{:}\neg\varphi \qquad s{:}\neg\psi}$$

$$\frac{s{:}\neg\neg\varphi}{s{:}\varphi}$$

$$\frac{s{:}\Diamond\varphi}{\begin{array}{c} sRt \\ t{:}\varphi \end{array}}$$

for t a new label

$$\frac{\begin{array}{c} s{:}\neg\Diamond\varphi \\ sRt \end{array}}{t{:}\neg\varphi}$$

$$\begin{array}{c}
s{:}(\Diamond p \wedge (\neg\Diamond\neg q \wedge \neg\Diamond(p \wedge q))) \\
s{:}\Diamond p \\
s{:}\neg\Diamond\neg q \wedge \neg\Diamond(p \wedge q) \\
s{:}\neg\Diamond\neg q \\
s{:}\neg\Diamond(p \wedge q) \\
sRt \\
t{:}p \\
t{:}\neg\neg q
\end{array}$$

The Complete Cast, plus an Example

$$\frac{s{:}(\varphi \wedge \psi)}{\begin{array}{c} s{:}\varphi \\ s{:}\psi \end{array}}$$

$$\frac{s{:}\neg(\varphi \wedge \psi)}{s{:}\neg\varphi \qquad s{:}\neg\psi}$$

$$\frac{s{:}\neg\neg\varphi}{s{:}\varphi}$$

$$\frac{s{:}\Diamond\varphi}{\begin{array}{c} sRt \\ t{:}\varphi \end{array}}$$

for t a new label

$$\frac{\begin{array}{c} s{:}\neg\Diamond\varphi \\ sRt \end{array}}{t{:}\neg\varphi}$$

$$\begin{array}{c}
s{:}(\Diamond p \wedge (\neg\Diamond\neg q \wedge \neg\Diamond(p \wedge q))) \\
s{:}\Diamond p \\
s{:}\neg\Diamond\neg q \wedge \neg\Diamond(p \wedge q) \\
s{:}\neg\Diamond\neg q \\
s{:}\neg\Diamond(p \wedge q) \\
sRt \\
t{:}p \\
t{:}\neg\neg q \\
t{:}q
\end{array}$$

The Complete Cast, plus an Example

$$\frac{s{:}(\varphi \wedge \psi)}{\begin{array}{c} s{:}\varphi \\ s{:}\psi \end{array}}$$

$$\frac{s{:}\neg(\varphi \wedge \psi)}{s{:}\neg\varphi \qquad s{:}\neg\psi}$$

$$\frac{s{:}\neg\neg\varphi}{s{:}\varphi}$$

$$\frac{s{:}\Diamond\varphi}{\begin{array}{c} sRt \\ t{:}\varphi \end{array}}$$

for t a new label

$$\frac{\begin{array}{c} s{:}\neg\Diamond\varphi \\ sRt \end{array}}{t{:}\neg\varphi}$$

$$\begin{array}{c}
s{:}(\Diamond p \wedge (\neg\Diamond\neg q \wedge \neg\Diamond(p \wedge q))) \\
s{:}\Diamond p \\
s{:}\neg\Diamond\neg q \wedge \neg\Diamond(p \wedge q) \\
s{:}\neg\Diamond\neg q \\
s{:}\neg\Diamond(p \wedge q) \\
sRt \\
t{:}p \\
t{:}\neg\neg q \\
t{:}q
\end{array}$$

The Complete Cast, plus an Example

$$\frac{s{:}(\varphi \wedge \psi)}{\begin{array}{c} s{:}\varphi \\ s{:}\psi \end{array}}$$

$$\frac{s{:}\neg(\varphi \wedge \psi)}{s{:}\neg\varphi \qquad s{:}\neg\psi}$$

$$\frac{s{:}\neg\neg\varphi}{s{:}\varphi}$$

$$\frac{s{:}\Diamond\varphi}{\begin{array}{c} sRt \\ t{:}\varphi \end{array}}$$

for t a new label

$$\frac{\begin{array}{c} s{:}\neg\Diamond\varphi \\ sRt \end{array}}{t{:}\neg\varphi}$$

$$\begin{array}{c}
s{:}(\Diamond p \wedge (\neg\Diamond\neg q \wedge \neg\Diamond(p \wedge q))) \\
s{:}\Diamond p \\
s{:}\neg\Diamond\neg q \wedge \neg\Diamond(p \wedge q) \\
s{:}\neg\Diamond\neg q \\
s{:}\neg\Diamond(p \wedge q) \\
sRt \\
t{:}p \\
t{:}\neg\neg q \\
t{:}q \\
t{:}\neg(p \wedge q)
\end{array}$$

The Complete Cast, plus an Example

$$\frac{s{:}(\varphi \wedge \psi)}{\begin{array}{c} s{:}\varphi \\ s{:}\psi \end{array}}$$

$$\frac{s{:}\neg(\varphi \wedge \psi)}{s{:}\neg\varphi \qquad s{:}\neg\psi}$$

$$\frac{s{:}\neg\neg\varphi}{s{:}\varphi}$$

$$\frac{s{:}\Diamond\varphi}{\begin{array}{c} sRt \\ t{:}\varphi \end{array}}$$

for t a new label

$$\frac{\begin{array}{c} s{:}\neg\Diamond\varphi \\ sRt \end{array}}{t{:}\neg\varphi}$$

$$\begin{array}{c}
s{:}(\Diamond p \wedge (\neg\Diamond\neg q \wedge \neg\Diamond(p \wedge q))) \\
s{:}\Diamond p \\
s{:}\neg\Diamond\neg q \wedge \neg\Diamond(p \wedge q) \\
s{:}\neg\Diamond\neg q \\
s{:}\neg\Diamond(p \wedge q) \\
sRt \\
t{:}p \\
t{:}\neg\neg q \\
t{:}q \\
t{:}\neg(p \wedge q)
\end{array}$$

The Complete Cast, plus an Example

$$\frac{s{:}(\varphi \land \psi)}{\begin{array}{c} s{:}\varphi \\ s{:}\psi \end{array}}$$

$$\frac{s{:}\neg(\varphi \land \psi)}{s{:}\neg\varphi \qquad\qquad s{:}\neg\psi}$$

$$\frac{s{:}\neg\neg\varphi}{s{:}\varphi}$$

$$\frac{s{:}\Diamond\varphi}{\begin{array}{c} sRt \\ t{:}\varphi \end{array}}$$

for t a new label

$$\frac{\begin{array}{c} s{:}\neg\Diamond\varphi \\ sRt \end{array}}{t{:}\neg\varphi}$$

$$s{:}(\Diamond p \land (\neg\Diamond\neg q \land \neg\Diamond(p \land q)))$$
$$s{:}\Diamond p$$
$$s{:}\neg\Diamond\neg q \land \neg\Diamond(p \land q)$$
$$s{:}\neg\Diamond\neg q$$
$$s{:}\neg\Diamond(p \land q)$$
$$sRt$$
$$t{:}p$$
$$t{:}\neg\neg q$$
$$t{:}q$$
$$t{:}\neg(p \land q)$$
$$t{:}\neg p \qquad\qquad t{:}\neg q$$

The Complete Cast, plus an Example

$$\frac{s{:}(\varphi \wedge \psi)}{\begin{array}{c} s{:}\varphi \\ s{:}\psi \end{array}}$$

$$\frac{s{:}\neg(\varphi \wedge \psi)}{s{:}\neg\varphi \qquad\qquad s{:}\neg\psi}$$

$$\frac{s{:}\neg\neg\varphi}{s{:}\varphi}$$

$$\frac{s{:}\Diamond\varphi}{\begin{array}{c} sRt \\ t{:}\varphi \end{array}}$$

for t a new label

$$\frac{\begin{array}{c} s{:}\neg\Diamond\varphi \\ sRt \end{array}}{t{:}\neg\varphi}$$

$$\begin{array}{c}
s{:}(\Diamond p \wedge (\neg\Diamond\neg q \wedge \neg\Diamond(p \wedge q))) \\
s{:}\Diamond p \\
s{:}\neg\Diamond\neg q \wedge \neg\Diamond(p \wedge q) \\
s{:}\neg\Diamond\neg q \\
s{:}\neg\Diamond(p \wedge q) \\
sRt \\
t{:}p \\
t{:}\neg\neg q \\
t{:}q \\
t{:}\neg(p \wedge q) \\
t{:}\neg p \qquad\qquad t{:}\neg q
\end{array}$$

closed closed

The Complete Cast, plus an Example

$$\frac{s{:}(\varphi \wedge \psi)}{\begin{array}{c} s{:}\varphi \\ s{:}\psi \end{array}} \qquad \frac{s{:}\Diamond\varphi}{\begin{array}{c} sRt \\ t{:}\varphi \end{array}}$$

for t a new label

$$\frac{s{:}\neg(\varphi \wedge \psi)}{s{:}\neg\varphi \qquad\qquad s{:}\neg\psi} \qquad \frac{\begin{array}{c} s{:}\neg\Diamond\varphi \\ sRt \end{array}}{t{:}\neg\varphi}$$

$$\frac{s{:}\neg\neg\varphi}{s{:}\varphi}$$

$$\begin{array}{c}
s{:}(\Diamond p \wedge (\neg\Diamond\neg q \wedge \neg\Diamond(p \wedge q))) \\
s{:}\Diamond p \\
s{:}\neg\Diamond\neg q \wedge \neg\Diamond(p \wedge q) \\
s{:}\neg\Diamond\neg q \\
s{:}\neg\Diamond(p \wedge q) \\
sRt \\
t{:}p \\
t{:}\neg\neg q \\
t{:}q \\
t{:}\neg(p \wedge q) \\
t{:}\neg p \qquad\qquad t{:}\neg q \\
\\
\text{closed} \qquad\qquad \text{closed}
\end{array}$$

- Which are the similarities/differences with tableaux for PL?

The Complete Cast, plus an Example

$$\frac{s{:}(\varphi \wedge \psi)}{\begin{array}{c} s{:}\varphi \\ s{:}\psi \end{array}}$$

$$\frac{s{:}\neg(\varphi \wedge \psi)}{s{:}\neg\varphi \qquad s{:}\neg\psi}$$

$$\frac{s{:}\neg\neg\varphi}{s{:}\varphi}$$

$$\frac{s{:}\Diamond\varphi}{\begin{array}{c} sRt \\ t{:}\varphi \end{array}}$$

for t a new label

$$\frac{\begin{array}{c} s{:}\neg\Diamond\varphi \\ sRt \end{array}}{t{:}\neg\varphi}$$

$$\begin{array}{c}
s{:}(\Diamond p \wedge (\neg\Diamond\neg q \wedge \neg\Diamond(p \wedge q))) \\
s{:}\Diamond p \\
s{:}\neg\Diamond\neg q \wedge \neg\Diamond(p \wedge q) \\
s{:}\neg\Diamond\neg q \\
s{:}\neg\Diamond(p \wedge q) \\
sRt \\
t{:}p \\
t{:}\neg\neg q \\
t{:}q \\
t{:}\neg(p \wedge q) \\
t{:}\neg p \qquad t{:}\neg q \\
\text{closed} \qquad \text{closed}
\end{array}$$

- Which are the similarities/differences with tableaux for PL?
- How do we know that we got it right?

The Complete Cast, plus an Example

$$\frac{s{:}(\varphi \wedge \psi)}{\begin{array}{c} s{:}\varphi \\ s{:}\psi \end{array}}$$

$$\frac{s{:}\Diamond\varphi}{\begin{array}{c} sRt \\ t{:}\varphi \end{array}}$$

for t a new label

$$\frac{s{:}\neg(\varphi \wedge \psi)}{s{:}\neg\varphi \qquad s{:}\neg\psi}$$

$$\frac{\begin{array}{c} s{:}\neg\Diamond\varphi \\ sRt \end{array}}{t{:}\neg\varphi}$$

$$\frac{s{:}\neg\neg\varphi}{s{:}\varphi}$$

$s{:}(\Diamond p \wedge (\neg\Diamond\neg q \wedge \neg\Diamond(p \wedge q)))$

$s{:}\Diamond p$

$s{:}\neg\Diamond\neg q \wedge \neg\Diamond(p \wedge q)$

$s{:}\neg\Diamond\neg q$

$s{:}\neg\Diamond(p \wedge q)$

sRt

$t{:}p$

$t{:}\neg\neg q$

$t{:}q$

$t{:}\neg(p \wedge q)$

$t{:}\neg p$ | $t{:}\neg q$

closed | closed

- Which are the similarities/differences with tableaux for PL?
- How do we know that we got it right?
- What can we learn from the calculus?

A Closer Look

A Closer Look

- Which similarities / differences with tableaux for PL?

$$\frac{s{:}(\varphi \wedge \psi)}{\begin{array}{c} s{:}\varphi \\ s{:}\psi \end{array}}$$

$$\frac{s{:}\neg(\varphi \wedge \psi)}{s{:}\neg\varphi \qquad s{:}\neg\psi}$$

$$\frac{s{:}\neg\neg\varphi}{s{:}\varphi}$$

$$\frac{s{:}\Diamond\varphi}{\begin{array}{c} sRt \\ t{:}\varphi \end{array}}$$

for t a new label

$$\frac{\begin{array}{c} s{:}\neg\Diamond\varphi \\ sRt \end{array}}{t{:}\neg\varphi}$$

A Closer Look

- Which similarities / differences with tableaux for PL?
  - What are labels? What are they doing? Can we use them?

$$\frac{s{:}(\varphi \wedge \psi)}{\begin{array}{c} s{:}\varphi \\ s{:}\psi \end{array}} \qquad \frac{s{:}\Diamond\varphi}{\begin{array}{c} sRt \\ t{:}\varphi \end{array}}$$

for t a new label

$$\frac{s{:}\neg(\varphi \wedge \psi)}{s{:}\neg\varphi \quad\quad s{:}\neg\psi} \qquad \frac{\begin{array}{c} s{:}\neg\Diamond\varphi \\ sRt \end{array}}{t{:}\neg\varphi}$$

$$\frac{s{:}\neg\neg\varphi}{s{:}\varphi}$$

A Closer Look

- Which similarities / differences with tableaux for PL?
  - What are labels? What are they doing? Can we use them?
  - Is this an algorithm?

$$\frac{s{:}(\varphi \wedge \psi)}{\begin{array}{c} s{:}\varphi \\ s{:}\psi \end{array}} \qquad \frac{s{:}\Diamond\varphi}{\begin{array}{c} sRt \\ t{:}\varphi \end{array}}$$

for t a new label

$$\frac{s{:}\neg(\varphi \wedge \psi)}{s{:}\neg\varphi \qquad s{:}\neg\psi} \qquad \frac{\begin{array}{c} s{:}\neg\Diamond\varphi \\ sRt \end{array}}{t{:}\neg\varphi}$$

$$\frac{s{:}\neg\neg\varphi}{s{:}\varphi}$$

A Closer Look

- Which similarities / differences with tableaux for PL?
  - What are labels? What are they doing? Can we use them?
  - Is this an algorithm?
  - Is it a good algorithm?

$$\frac{s{:}(\varphi \wedge \psi)}{\begin{array}{c} s{:}\varphi \\ s{:}\psi \end{array}} \qquad \frac{s{:}\Diamond\varphi}{\begin{array}{c} sRt \\ t{:}\varphi \end{array}} \text{ for } t \text{ a new label}$$

$$\frac{s{:}\neg(\varphi \wedge \psi)}{s{:}\neg\varphi \quad \mid \quad s{:}\neg\psi} \qquad \frac{\begin{array}{c} s{:}\neg\Diamond\varphi \\ sRt \end{array}}{t{:}\neg\varphi}$$

$$\frac{s{:}\neg\neg\varphi}{s{:}\varphi}$$

A Closer Look

- Which similarities / differences with tableaux for PL?
  - What are labels? What are they doing? Can we use them?
  - Is this an algorithm?
  - Is it a good algorithm?
  - Does it terminate?

$$\frac{s{:}(\varphi \wedge \psi)}{\begin{array}{c} s{:}\varphi \\ s{:}\psi \end{array}}$$

$$\frac{s{:}\neg(\varphi \wedge \psi)}{s{:}\neg\varphi \qquad s{:}\neg\psi}$$

$$\frac{s{:}\neg\neg\varphi}{s{:}\varphi}$$

$$\frac{s{:}\Diamond\varphi}{\begin{array}{c} sRt \\ t{:}\varphi \end{array}}$$

for t a new label

$$\frac{\begin{array}{c} s{:}\neg\Diamond\varphi \\ sRt \end{array}}{t{:}\neg\varphi}$$

A Closer Look

- Which similarities / differences with tableaux for PL?
  - What are labels? What are they doing? Can we use them?
  - Is this an algorithm?
  - Is it a good algorithm?
  - Does it terminate?
- Did we get it right?

$$\frac{s{:}(\varphi \wedge \psi)}{\begin{array}{c} s{:}\varphi \\ s{:}\psi \end{array}} \qquad \frac{s{:}\Diamond\varphi}{\begin{array}{c} sRt \\ t{:}\varphi \end{array}}$$

for t a new label

$$\frac{s{:}\neg(\varphi \wedge \psi)}{s{:}\neg\varphi \quad \mid \quad s{:}\neg\psi} \qquad \frac{\begin{array}{c} s{:}\neg\Diamond\varphi \\ sRt \end{array}}{t{:}\neg\varphi}$$

$$\frac{s{:}\neg\neg\varphi}{s{:}\varphi}$$

A Closer Look

- Which similarities / differences with tableaux for PL?
  - What are labels? What are they doing? Can we use them?
  - Is this an algorithm?
  - Is it a good algorithm?
  - Does it terminate?

$$\frac{s{:}(\varphi \wedge \psi)}{\begin{array}{c} s{:}\varphi \\ s{:}\psi \end{array}} \qquad \frac{s{:}\Diamond\varphi}{\begin{array}{c} sRt \\ t{:}\varphi \end{array}} \text{ for } t \text{ a new label}$$

$$\frac{s{:}\neg(\varphi \wedge \psi)}{s{:}\neg\varphi \quad \mid \quad s{:}\neg\psi} \qquad \frac{\begin{array}{c} s{:}\neg\Diamond\varphi \\ sRt \end{array}}{t{:}\neg\varphi}$$

$$\frac{s{:}\neg\neg\varphi}{s{:}\varphi}$$

- Did we get it right?
  - Did we get it right in the PL case, to start with?! Consider the rule:

$$\frac{s{:}\neg(\varphi \wedge \psi)}{s{:}\neg\varphi \quad \mid \quad \begin{array}{c} s{:}\varphi \\ s{:}\neg\psi \end{array}}$$

A Closer Look

- Which similarities / differences with tableaux for PL?
  - What are labels? What are they doing? Can we use them?
  - Is this an algorithm?
  - Is it a good algorithm?
  - Does it terminate?
- Did we get it right?
  - Did we get it right in the PL case, to start with?! Consider the rule:
  - We should prove soundness and completeness

$$\frac{s{:}(\varphi \wedge \psi)}{\begin{array}{c} s{:}\varphi \\ s{:}\psi \end{array}} \qquad \frac{s{:}\Diamond\varphi}{\begin{array}{c} sRt \\ t{:}\varphi \end{array}} \text{ for } t \text{ a new label}$$

$$\frac{s{:}\neg(\varphi \wedge \psi)}{s{:}\neg\varphi \quad \mid \quad s{:}\neg\psi} \qquad \frac{\begin{array}{c} s{:}\neg\Diamond\varphi \\ sRt \end{array}}{t{:}\neg\varphi}$$

$$\frac{s{:}\neg\neg\varphi}{s{:}\varphi}$$

$$\frac{s{:}\neg(\varphi \wedge \psi)}{s{:}\neg\varphi \quad \mid \quad \begin{array}{c} s{:}\varphi \\ s{:}\neg\psi \end{array}}$$

A Closer Look

- Which similarities / differences with tableaux for PL?
  - What are labels? What are they doing? Can we use them?
  - Is this an algorithm?
  - Is it a good algorithm?
  - Does it terminate?
- Did we get it right?
  - Did we get it right in the PL case, to start with?! Consider the rule:
  - We should prove soundness and completeness
- What can we learn from the calculus?

$$\frac{s{:}(\varphi \wedge \psi)}{\begin{array}{c} s{:}\varphi \\ s{:}\psi \end{array}} \qquad \frac{s{:}\Diamond\varphi}{\begin{array}{c} sRt \\ t{:}\varphi \end{array}} \text{ for } t \text{ a new label}$$

$$\frac{s{:}\neg(\varphi \wedge \psi)}{s{:}\neg\varphi \quad \mid \quad s{:}\neg\psi} \qquad \frac{\begin{array}{c} s{:}\neg\Diamond\varphi \\ sRt \end{array}}{t{:}\neg\varphi}$$

$$\frac{s{:}\neg\neg\varphi}{s{:}\varphi}$$

$$\frac{s{:}\neg(\varphi \wedge \psi)}{s{:}\neg\varphi \quad \mid \quad \begin{array}{c} s{:}\varphi \\ s{:}\neg\psi \end{array}}$$

A Closer Look

- Which similarities / differences with tableaux for PL?
  - What are labels? What are they doing? Can we use them?
  - Is this an algorithm?
  - Is it a good algorithm?
  - Does it terminate?

$$\frac{s{:}(\varphi \wedge \psi)}{\begin{array}{c} s{:}\varphi \\ s{:}\psi \end{array}} \qquad \frac{s{:}\Diamond\varphi}{\begin{array}{c} sRt \\ t{:}\varphi \end{array}} \text{ for } t \text{ a new label}$$

$$\frac{s{:}\neg(\varphi \wedge \psi)}{s{:}\neg\varphi \quad \mid \quad s{:}\neg\psi} \qquad \frac{\begin{array}{c} s{:}\neg\Diamond\varphi \\ sRt \end{array}}{t{:}\neg\varphi}$$

$$\frac{s{:}\neg\neg\varphi}{s{:}\varphi}$$

- Did we get it right?
  - Did we get it right in the PL case, to start with?! Consider the rule:

$$\frac{s{:}\neg(\varphi \wedge \psi)}{s{:}\neg\varphi \quad \mid \quad \begin{array}{c} s{:}\varphi \\ s{:}\neg\psi \end{array}}$$

  - We should prove soundness and completeness
- What can we learn from the calculus?
  - Something about models!

Tree Models

Tree Models

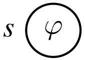
- Let us see the tableux proof we did before again, for the formula

$$\varphi = s{:}(\Diamond p \land (\neg\Diamond\neg q \land \neg\Diamond(p \land q)))$$

Tree Models

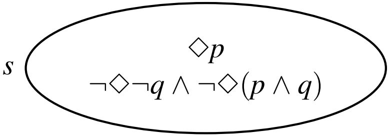
- Let us see the tableux proof we did before again, for the formula

$$\varphi = s{:}(\Diamond p \wedge (\neg\Diamond\neg q \wedge \neg\Diamond(p \wedge q)))$$



Tree Models

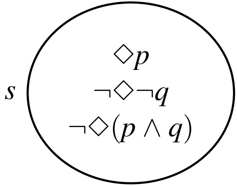
- Let us see the tableux proof we did before again, for the formula



$$\varphi = s{:}(\Diamond p \wedge (\neg\Diamond\neg q \wedge \neg\Diamond(p \wedge q)))$$
$$s{:}\Diamond p$$
$$s{:}\neg\Diamond\neg q \wedge \neg\Diamond(p \wedge q)$$

Tree Models

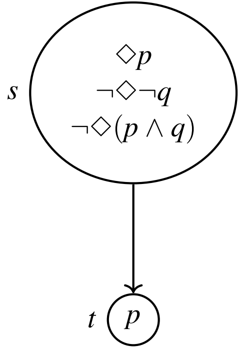
- Let us see the tableux proof we did before again, for the formula



$$\varphi = s{:}(\Diamond p \wedge (\neg\Diamond\neg q \wedge \neg\Diamond(p \wedge q)))$$
$$s{:}\Diamond p$$
$$s{:}\neg\Diamond\neg q \wedge \neg\Diamond(p \wedge q)$$
$$s{:}\neg\Diamond\neg q$$
$$s{:}\neg\Diamond(p \wedge q)$$

Tree Models

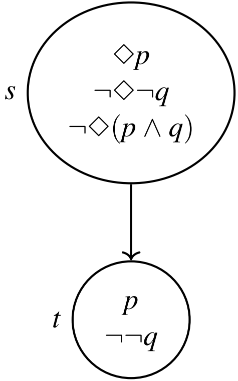
- Let us see the tableux proof we did before again, for the formula



$$\varphi = s{:}(\Diamond p \wedge (\neg\Diamond\neg q \wedge \neg\Diamond(p \wedge q)))$$
$$s{:}\Diamond p$$
$$s{:}\neg\Diamond\neg q \wedge \neg\Diamond(p \wedge q)$$
$$s{:}\neg\Diamond\neg q$$
$$s{:}\neg\Diamond(p \wedge q)$$
$$sRt$$
$$t{:}p$$

Tree Models

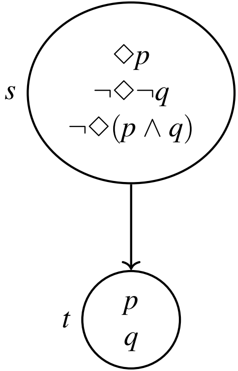
- Let us see the tableux proof we did before again, for the formula



$$\varphi = s{:}(\Diamond p \wedge (\neg\Diamond\neg q \wedge \neg\Diamond(p \wedge q)))$$
$$s{:}\Diamond p$$
$$s{:}\neg\Diamond\neg q \wedge \neg\Diamond(p \wedge q)$$
$$s{:}\neg\Diamond\neg q$$
$$s{:}\neg\Diamond(p \wedge q)$$
$$sRt$$
$$t{:}p$$
$$t{:}\neg\neg q$$

Tree Models

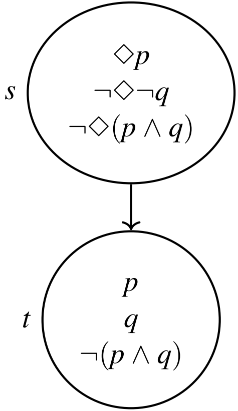
- Let us see the tableux proof we did before again, for the formula



$$\varphi = s{:}(\Diamond p \wedge (\neg\Diamond\neg q \wedge \neg\Diamond(p \wedge q)))$$
$$s{:}\Diamond p$$
$$s{:}\neg\Diamond\neg q \wedge \neg\Diamond(p \wedge q)$$
$$s{:}\neg\Diamond\neg q$$
$$s{:}\neg\Diamond(p \wedge q)$$
$$sRt$$
$$t{:}p$$
$$t{:}\neg\neg q$$
$$t{:}q$$

Tree Models

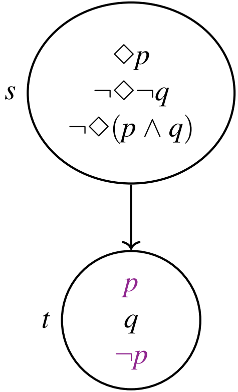
- Let us see the tableux proof we did before again, for the formula



$$\varphi = s{:}(\Diamond p \wedge (\neg\Diamond\neg q \wedge \neg\Diamond(p \wedge q)))$$
$$s{:}\Diamond p$$
$$s{:}\neg\Diamond\neg q \wedge \neg\Diamond(p \wedge q)$$
$$s{:}\neg\Diamond\neg q$$
$$s{:}\neg\Diamond(p \wedge q)$$
$$sRt$$
$$t{:}p$$
$$t{:}\neg\neg q$$
$$t{:}q$$
$$t{:}\neg(p \wedge q)$$

Tree Models

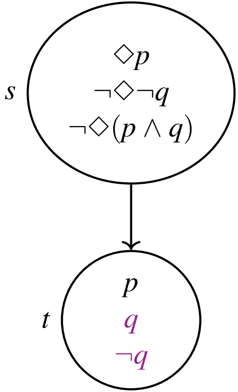
- Let us see the tableux proof we did before again, for the formula



$$\varphi = s{:}(\Diamond p \wedge (\neg\Diamond\neg q \wedge \neg\Diamond(p \wedge q)))$$
$$s{:}\Diamond p$$
$$s{:}\neg\Diamond\neg q \wedge \neg\Diamond(p \wedge q)$$
$$s{:}\neg\Diamond\neg q$$
$$s{:}\neg\Diamond(p \wedge q)$$
$$sRt$$
$$t{:}p$$
$$t{:}\neg\neg q$$
$$t{:}q$$
$$t{:}\neg(p \wedge q)$$
$$t{:}\neg p \qquad\qquad t{:}\neg q$$

Tree Models

- Let us see the tableux proof we did before again, for the formula



$$\varphi = s{:}(\Diamond p \wedge (\neg\Diamond\neg q \wedge \neg\Diamond(p \wedge q)))$$
$$s{:}\Diamond p$$
$$s{:}\neg\Diamond\neg q \wedge \neg\Diamond(p \wedge q)$$
$$s{:}\neg\Diamond\neg q$$
$$s{:}\neg\Diamond(p \wedge q)$$
$$sRt$$
$$t{:}p$$
$$t{:}\neg\neg q$$
$$t{:}q$$
$$t{:}\neg(p \wedge q)$$
$$t{:}\neg p \qquad\qquad t{:}\neg q$$

Tree and Finite Model Properties

Tree and Finite Model Properties

- Using the rules of the tableaux calculus we only explore finite, tree models.

$$\frac{s{:}(\varphi \wedge \psi)}{\begin{array}{c} s{:}\varphi \\ s{:}\psi \end{array}} \qquad \frac{s{:}\Diamond\varphi}{\begin{array}{c} sRt \\ t{:}\varphi \end{array}}$$

for t a new label

$$\frac{s{:}\neg(\varphi \wedge \psi)}{s{:}\neg\varphi \quad \mid \quad s{:}\neg\psi} \qquad \frac{\begin{array}{c} s{:}\neg\Diamond\varphi \\ sRt \end{array}}{t{:}\neg\varphi}$$

$$\frac{s{:}\neg\neg\varphi}{s{:}\varphi}$$

Tree and Finite Model Properties

- Using the rules of the tableaux calculus we only explore finite, tree models.
- Let's assume that the calculus is correct.

$$\frac{s{:}(\varphi \wedge \psi)}{\begin{array}{c} s{:}\varphi \\ s{:}\psi \end{array}} \qquad \frac{s{:}\Diamond\varphi}{\begin{array}{c} sRt \\ t{:}\varphi \end{array}}$$

for t a new label

$$\frac{s{:}\neg(\varphi \wedge \psi)}{s{:}\neg\varphi \quad \mid \quad s{:}\neg\psi} \qquad \frac{\begin{array}{c} s{:}\neg\Diamond\varphi \\ sRt \end{array}}{t{:}\neg\varphi}$$

$$\frac{s{:}\neg\neg\varphi}{s{:}\varphi}$$

Tree and Finite Model Properties

- Using the rules of the tableaux calculus we only explore finite, tree models.
- Let's assume that the calculus is correct.
- Then BML
  - cannot say infinite,
  - cannot say non-tree.

$$\frac{s{:}(\varphi \wedge \psi)}{\begin{array}{c} s{:}\varphi \\ s{:}\psi \end{array}} \qquad \frac{s{:}\Diamond\varphi}{\begin{array}{c} sRt \\ t{:}\varphi \end{array}}$$

for t a new label

$$\frac{s{:}\neg(\varphi \wedge \psi)}{s{:}\neg\varphi \quad \mid \quad s{:}\neg\psi} \qquad \frac{\begin{array}{c} s{:}\neg\Diamond\varphi \\ sRt \end{array}}{t{:}\neg\varphi}$$

$$\frac{s{:}\neg\neg\varphi}{s{:}\varphi}$$

Tree and Finite Model Properties

- Using the rules of the tableaux calculus we only explore finite, tree models.
- Let's assume that the calculus is correct.
- Then BML
  - cannot say infinite,
  - cannot say non-tree.

$$\frac{s{:}(\varphi \wedge \psi)}{\begin{array}{c} s{:}\varphi \\ s{:}\psi \end{array}} \qquad \frac{s{:}\Diamond\varphi}{\begin{array}{c} sRt \\ t{:}\varphi \end{array}}$$

for t a new label

$$\frac{s{:}\neg(\varphi \wedge \psi)}{s{:}\neg\varphi \quad\quad s{:}\neg\psi} \qquad \frac{\begin{array}{c} s{:}\neg\Diamond\varphi \\ sRt \end{array}}{t{:}\neg\varphi}$$

$$\frac{s{:}\neg\neg\varphi}{s{:}\varphi}$$

Theorem: A formula in the $\langle R\rangle$-language is satisfialle if and only if it is satisfiable in a finite, tree relational structure.

Soundness and Completeness

- We would like to verify that:
  - if the tableau for φ is closed (all tranches contains a class) then φ is UNSAT [Soundness].
  - if the tableau for φ cannot be further extended and it has an open branch then φ is SAT [Completeness].

Soundness and Completeness

- We would like to verify that:
  - if the tableau for φ is closed (all tranches contains a class) then φ is UNSAT [Soundness].
  - if the tableau for φ cannot be further extended and it has an open branch then φ is SAT [Completeness].
- Soundness is usually easy to establish. Prove, for each rule of the tableaux, that if the antecedent has a model, then at least one of the generated branches has a model.

Soundness and Completeness

- We would like to verify that:
  - if the tableau for φ is closed (all tranches contains a class) then φ is UNSAT [Soundness].
  - if the tableau for φ cannot be further extended and it has an open branch then φ is SAT [Completeness].
- Soundness is usually easy to establish. Prove, for each rule of the tableaux, that if the antecedent has a model, then at least one of the generated branches has a model.
- To show completeness we need to build a model from a saturated, open branch.

Completeness

Theorem. If Γ is a saturated open branch from a tableaux for φ, then φ is SAT.

Proof. Given Γ we define the model $\mathcal{M}_\Gamma = \langle W_\Gamma, R_\Gamma, V_\Gamma \rangle$ where

$$
\begin{array}{rcl}
W_\Gamma & = & \{w \mid w{:}\varphi \in \Gamma\} \\
R_\Gamma & = & \{(w, v) \mid wRv \in \Gamma\} \\
V_\Gamma(p) & = & \{w \mid w{:}p \in \Gamma\}
\end{array}
$$

Completeness

Theorem. If Γ is a saturated open branch from a tableaux for φ, then φ is SAT.

Proof. Given Γ we define the model $\mathcal{M}_\Gamma = \langle W_\Gamma, R_\Gamma, V_\Gamma \rangle$ where

$$\begin{array}{rcl} W_\Gamma & = & \{w \mid w{:}\varphi \in \Gamma\} \\ R_\Gamma & = & \{(w, v) \mid wRv \in \Gamma\} \\ V_\Gamma(p) & = & \{w \mid w{:}p \in \Gamma\} \end{array}$$

Let ψ be the smallest formula such that $w{:}\psi \in \Gamma$ and $\mathcal{M}_\Gamma, w \not\models \psi$.

Completeness

Theorem. If Γ is a saturated open branch from a tableaux for φ, then φ is SAT.

Proof. Given Γ we define the model $\mathcal{M}_\Gamma = \langle W_\Gamma, R_\Gamma, V_\Gamma \rangle$ where

$$\begin{array}{rcl} W_\Gamma & = & \{w \mid w{:}\varphi \in \Gamma\} \\ R_\Gamma & = & \{(w,v) \mid wRv \in \Gamma\} \\ V_\Gamma(p) & = & \{w \mid w{:}p \in \Gamma\} \end{array}$$

Let ψ be the smallest formula such that $w{:}\psi \in \Gamma$ and $\mathcal{M}_\Gamma, w \not\models \psi$.

- $\psi \neq p$ (otherwise $w \in V_\Gamma(p)$) and $\psi \neq \neg p$ (the branch would be closed).

Completeness

Theorem. If Γ is a saturated open branch from a tableaux for φ, then φ is SAT.

Proof. Given Γ we define the model $\mathcal{M}_\Gamma = \langle W_\Gamma, R_\Gamma, V_\Gamma \rangle$ where

$$\begin{array}{rcl} W_\Gamma & = & \{w \mid w{:}\varphi \in \Gamma\} \\ R_\Gamma & = & \{(w, v) \mid wRv \in \Gamma\} \\ V_\Gamma(p) & = & \{w \mid w{:}p \in \Gamma\} \end{array}$$

Let ψ be the smallest formula such that $w{:}\psi \in \Gamma$ and $\mathcal{M}_\Gamma, w \not\models \psi$.

- $\psi \neq p$ (otherwise $w \in V_\Gamma(p)$) and $\psi \neq \neg p$ (the branch would be closed).
- $\psi \neq \psi_1 \wedge \psi_2$ otherwise as both $\psi_i \in \Gamma$, ψ won't be minimal.

Completeness

Theorem. If Γ is a saturated open branch from a tableaux for φ, then φ is SAT.

Proof. Given Γ we define the model $\mathcal{M}_\Gamma = \langle W_\Gamma, R_\Gamma, V_\Gamma \rangle$ where

$$\begin{array}{rcl} W_\Gamma & = & \{w \mid w{:}\varphi \in \Gamma\} \\ R_\Gamma & = & \{(w,v) \mid wRv \in \Gamma\} \\ V_\Gamma(p) & = & \{w \mid w{:}p \in \Gamma\} \end{array}$$

Let ψ be the smallest formula such that $w{:}\psi \in \Gamma$ and $\mathcal{M}_\Gamma, w \not\models \psi$.

- $\psi \neq p$ (otherwise $w \in V_\Gamma(p)$) and $\psi \neq \neg p$ (the branch would be closed).
- $\psi \neq \psi_1 \wedge \psi_2$ otherwise as both $\psi_i \in \Gamma$, ψ won't be minimal.
- $\psi \neq \neg(\psi_1 \wedge \psi_2)$ for similar reasons.

Completeness

Theorem. If Γ is a saturated open branch from a tableaux for φ, then φ is SAT.

Proof. Given Γ we define the model $\mathcal{M}_\Gamma = \langle W_\Gamma, R_\Gamma, V_\Gamma \rangle$ where

$$\begin{aligned} W_\Gamma &= \{w \mid w{:}\varphi \in \Gamma\} \\ R_\Gamma &= \{(w,v) \mid wRv \in \Gamma\} \\ V_\Gamma(p) &= \{w \mid w{:}p \in \Gamma\} \end{aligned}$$

Let ψ be the smallest formula such that $w{:}\psi \in \Gamma$ and $\mathcal{M}_\Gamma, w \not\models \psi$.

- $\psi \neq p$ (otherwise $w \in V_\Gamma(p)$) and $\psi \neq \neg p$ (the branch would be closed).
- $\psi \neq \psi_1 \wedge \psi_2$ otherwise as both $\psi_i \in \Gamma$, ψ won't be minimal.
- $\psi \neq \neg(\psi_1 \wedge \psi_2)$ for similar reasons.
- $\psi \neq \langle R \rangle \xi$ as we would have wRv and $v{:}\xi \in \Gamma$ and ψ won't be minimal.

Completeness

Theorem. If Γ is a saturated open branch from a tableaux for φ, then φ is SAT.

Proof. Given Γ we define the model $\mathcal{M}_\Gamma = \langle W_\Gamma, R_\Gamma, V_\Gamma \rangle$ where

$$\begin{array}{rcl} W_\Gamma & = & \{w \mid w{:}\varphi \in \Gamma\} \\ R_\Gamma & = & \{(w,v) \mid wRv \in \Gamma\} \\ V_\Gamma(p) & = & \{w \mid w{:}p \in \Gamma\} \end{array}$$

Let ψ be the smallest formula such that $w{:}\psi \in \Gamma$ and $\mathcal{M}_\Gamma, w \not\models \psi$.

- $\psi \neq p$ (otherwise $w \in V_\Gamma(p)$) and $\psi \neq \neg p$ (the branch would be closed).
- $\psi \neq \psi_1 \wedge \psi_2$ otherwise as both $\psi_i \in \Gamma$, ψ won't be minimal.
- $\psi \neq \neg(\psi_1 \wedge \psi_2)$ for similar reasons.
- $\psi \neq \langle R \rangle \xi$ as we would have wRv and $v{:}\xi \in \Gamma$ and ψ won't be minimal.
- Similarly for $\neg R\xi$.

Completeness

Theorem. If Γ is a saturated open branch from a tableaux for φ, then φ is SAT.

Proof. Given Γ we define the model $\mathcal{M}_\Gamma = \langle W_\Gamma, R_\Gamma, V_\Gamma \rangle$ where

$$\begin{array}{rcl} W_\Gamma & = & \{w \mid w{:}\varphi \in \Gamma\} \\ R_\Gamma & = & \{(w,v) \mid wRv \in \Gamma\} \\ V_\Gamma(p) & = & \{w \mid w{:}p \in \Gamma\} \end{array}$$

Let ψ be the smallest formula such that $w{:}\psi \in \Gamma$ and $\mathcal{M}_\Gamma, w \not\models \psi$.

- $\psi \neq p$ (otherwise $w \in V_\Gamma(p)$) and $\psi \neq \neg p$ (the branch would be closed).
- $\psi \neq \psi_1 \wedge \psi_2$ otherwise as both $\psi_i \in \Gamma$, ψ won't be minimal.
- $\psi \neq \neg(\psi_1 \wedge \psi_2)$ for similar reasons.
- $\psi \neq \langle R \rangle \xi$ as we would have wRv and $v{:}\xi \in \Gamma$ and ψ won't be minimal.
- Similarly for $\neg R\xi$.

Hence, $w{:}\varphi \in \Gamma$ implies $\mathcal{M}_\Gamma, w \models \varphi$.

What we want to cover

- Introduce the basic hybrid logics $\mathcal{H}(@)$.
- Extend the standard translation to $\mathcal{H}(@)$.
- Talk about expressive power. Bisimulation.
- Define a tableax algorithm for satisfiability of formulas in the basic modal logic (BML).
  - How do we check whether a formula has a model?
  - What can we learn from tableaux?
- Transform the tableau for BML into one for $\mathcal{H}(@)$.
- Discuss more expressive hybrid logics.

But what about Hybrid Logics?

- We devised a sound and complete (and terminating) tableau for the basic modal logic.

But what about Hybrid Logics?

- We devised a sound and complete (and terminating) tableau for the basic modal logic. But this is a lecture about Hybrid Logics!!!

But what about Hybrid Logics?

- We devised a sound and complete (and terminating) tableau for the basic modal logic. But this is a lecture about Hybrid Logics!!!
- We have been using hybrid logics all the time
  - Write $@_s\varphi$ instead of $s{:}\varphi$
  - Write $@_s\langle R\rangle t$ instead of sRt

But what about Hybrid Logics?

- We devised a sound and complete (and terminating) tableau for the basic modal logic. But this is a lecture about Hybrid Logics!!!
- We have been using hybrid logics all the time
  - Write $@_s\varphi$ instead of $s{:}\varphi$
  - Write $@_s\langle R\rangle t$ instead of sRt

$$\frac{@_s(\varphi \wedge \psi)}{\begin{array}{c}@_s\varphi\\ @_s\psi\end{array}} \qquad \frac{@_s\Diamond\varphi}{\begin{array}{c}@_s\langle R\rangle t\\ @_t\varphi\end{array}}$$

for t a new label

$$\frac{@_s\neg(\varphi \wedge \psi)}{@_s\neg\varphi \qquad\qquad @_s\neg\psi} \qquad \frac{\begin{array}{c}@_s\neg\Diamond\varphi\\ @_s\langle R\rangle t\end{array}}{@_t\neg\varphi}$$

$$\frac{@_s\neg\neg\varphi}{@_s\varphi}$$

But what about Hybrid Logics?

- We devised a sound and complete (and terminating) tableau for the basic modal logic. But this is a lecture about Hybrid Logics!!!
- We have been using hybrid logics all the time
  - Write $@_s\varphi$ instead of $s{:}\varphi$
  - Write $@_s\langle R\rangle t$ instead of sRt
- But the set of rules is not complete for the whole basic hybrid language

$$\frac{@_s(\varphi \wedge \psi)}{\begin{array}{c}@_s\varphi\\ @_s\psi\end{array}} \qquad \frac{@_s\Diamond\varphi}{\begin{array}{c}@_s\langle R\rangle t\\ @_t\varphi\end{array}} \text{ for } t \text{ a new label}$$

$$\frac{@_s\neg(\varphi \wedge \psi)}{@_s\neg\varphi \quad \mid \quad @_s\neg\psi} \qquad \frac{\begin{array}{c}@_s\neg\Diamond\varphi\\ @_s\langle R\rangle t\end{array}}{@_t\neg\varphi}$$

$$\frac{@_s\neg\neg\varphi}{@_s\varphi}$$

@-Formulas

- Notation: For simplicity and to keep the notation we were using we will keep writing $i{:}\varphi$ instead of $@_i\varphi$.

@-Formulas

- Notation: For simplicity and to keep the notation we were using we will keep writing $i{:}\varphi$ instead of $@_i\varphi$.
- Fact 1: φ is satisfiable iff $i{:}\varphi$ is satisfiable, for i not in φ.

@-Formulas

- Notation: For simplicity and to keep the notation we were using we will keep writing $i{:}\varphi$ instead of $@_i\varphi$.
- Fact 1: φ is satisfiable iff $i{:}\varphi$ is satisfiable, for i not in φ. I.e., to check the satisfiability of φ we start the tableax with $i{:}\varphi$.

@-Formulas

- Notation: For simplicity and to keep the notation we were using we will keep writing $i{:}\varphi$ instead of $@_i\varphi$.
- Fact 1: φ is satisfiable iff $i{:}\varphi$ is satisfiable, for i not in φ. I.e., to check the satisfiability of φ we start the tableax with $i{:}\varphi$.
- Fact 2: : is self-dual, hence $\neg i{:}\varphi$ is equivalent to $i{:}\neg\varphi$.

Sound and Complete Tableaux for $\mathcal{H}(@)$

Sound and Complete Tableaux for $\mathcal{H}(@)$

$\neg$ rules:

$$(\neg)\ \frac{i{:}\neg\neg\varphi}{i{:}\varphi}$$

Sound and Complete Tableaux for $\mathcal{H}(@)$

$\neg$ rules: $$(\neg)\ \frac{i{:}\neg\neg\varphi}{i{:}\varphi}$$

$\wedge$ rules: $$(\wedge)\ \frac{i{:}(\varphi \wedge \psi)}{\begin{array}{c} i{:}\varphi \\ i{:}\psi \end{array}} \qquad (\neg\wedge)\ \frac{i{:}\neg(\varphi \wedge \psi)}{i{:}\neg\varphi \quad s{:}\neg\psi}$$

Sound and Complete Tableaux for $\mathcal{H}(@)$

$\neg$ rules:

$$(\neg)\ \frac{i{:}\neg\neg\varphi}{i{:}\varphi}$$

$\wedge$ rules:

$$(\wedge)\ \frac{i{:}(\varphi \wedge \psi)}{\begin{array}{c} i{:}\varphi \\ i{:}\psi \end{array}} \qquad (\neg\wedge)\ \frac{i{:}\neg(\varphi \wedge \psi)}{i{:}\neg\varphi \quad s{:}\neg\psi}$$

$\langle r \rangle$ rules:

$$(\langle r \rangle)\ \frac{i{:}\langle r \rangle\varphi}{\begin{array}{c} i{:}\langle r \rangle j \\ j{:}\varphi \end{array}} \qquad (\neg\langle r \rangle)\ \frac{\begin{array}{c} i{:}\neg\langle r \rangle\varphi \\ i{:}\langle r \rangle j \end{array}}{j{:}\neg\varphi}$$

for j new in branch

Sound and Complete Tableaux for $\mathcal{H}(@)$

$\neg$ rules: $(\neg)\ \dfrac{i{:}\neg\neg\varphi}{i{:}\varphi}$

$\wedge$ rules: $(\wedge)\ \dfrac{i{:}(\varphi \wedge \psi)}{\begin{array}{c} i{:}\varphi \\ i{:}\psi \end{array}}$ $\qquad (\neg\wedge)\ \dfrac{i{:}\neg(\varphi \wedge \psi)}{i{:}\neg\varphi \quad s{:}\neg\psi}$

$\langle r\rangle$ rules: $(\langle r\rangle)\ \dfrac{i{:}\langle r\rangle\varphi}{\begin{array}{c} i{:}\langle r\rangle j \\ j{:}\varphi \end{array}}$ for j new in branch $\qquad (\neg\langle r\rangle)\ \dfrac{\begin{array}{c} i{:}\neg\langle r\rangle\varphi \\ i{:}\langle r\rangle j \end{array}}{j{:}\neg\varphi}$

@ rules: $(@)\ \dfrac{i{:}j{:}\varphi}{j{:}\varphi}$ $\qquad (\neg @)\ \dfrac{i{:}\neg j{:}\varphi}{j{:}\neg\varphi}$

Sound and Complete Tableaux for $\mathcal{H}(@)$

$\neg$ rules: $(\neg)\ \dfrac{i{:}\neg\neg\varphi}{i{:}\varphi}$

$\wedge$ rules: $(\wedge)\ \dfrac{i{:}(\varphi \wedge \psi)}{\begin{array}{c} i{:}\varphi \\ i{:}\psi \end{array}}$ $\qquad (\neg\wedge)\ \dfrac{i{:}\neg(\varphi \wedge \psi)}{i{:}\neg\varphi \quad s{:}\neg\psi}$

$\langle r \rangle$ rules: $(\langle r \rangle)\ \dfrac{i{:}\langle r \rangle\varphi}{\begin{array}{c} i{:}\langle r \rangle j \\ j{:}\varphi \end{array}}$ for j new in branch $\qquad (\neg\langle r \rangle)\ \dfrac{\begin{array}{c} i{:}\neg\langle r \rangle\varphi \\ i{:}\langle r \rangle j \end{array}}{j{:}\neg\varphi}$

@ rules: $(@)\ \dfrac{i{:}j{:}\varphi}{j{:}\varphi}$ $\qquad (\neg @)\ \dfrac{i{:}\neg j{:}\varphi}{j{:}\neg\varphi}$

Equality: (Ref) $\dfrac{[i \text{ on branch}]}{i{:}i}$ $\qquad$ (Sym) $\dfrac{i{:}j}{j{:}i}$ $\qquad$ (Cong) $\dfrac{i{:}k \quad j{:}k \quad i{:}\varphi}{j{:}\varphi}$

Hybrid Termination

- Once nominals and satisfaction operators are introduced, ensuring termination is more difficult.

Hybrid Termination

- Once nominals and satisfaction operators are introduced, ensuring termination is more difficult.
- An obvious problem is the *Cong* rule: $\dfrac{i{:}k \quad j{:}k \quad i{:}\varphi}{j{:}\varphi}$

Hybrid Termination

- Once nominals and satisfaction operators are introduced, ensuring termination is more difficult.
- An obvious problem is the *Cong* rule: $\dfrac{i{:}k \quad j{:}k \quad i{:}\varphi}{j{:}\varphi}$
- The solution is to impose a "direction": Only one nominal in the equivalence class is saturated. But equality should still be an equivalence, so an irrestricted version for nominals is introduced.

$$(\text{wCong}) \; \frac{i{:}k \quad j{:}k \quad i{:}\varphi}{j{:}\varphi}$$

j is the earliest introduced nominal making *k* true

Hybrid Termination

- Once nominals and satisfaction operators are introduced, ensuring termination is more difficult.
- An obvious problem is the *Cong* rule: $\dfrac{i{:}k \quad j{:}k \quad i{:}\varphi}{j{:}\varphi}$
- The solution is to impose a "direction": Only one nominal in the equivalence class is saturated. But equality should still be an equivalence, so an irrestricted version for nominals is introduced.

$$\text{(wCong)}\ \frac{i{:}k \quad j{:}k \quad i{:}\varphi}{j{:}\varphi}\ \begin{array}{c}j \text{ is the earliest introduced}\\ \text{nominal making } k \text{ true}\end{array} \qquad \text{(Nom)}\ \frac{i{:}k \quad j{:}k \quad i{:}n}{j{:}n}$$

Hybrid Termination

- Once nominals and satisfaction operators are introduced, ensuring termination is more difficult.
- An obvious problem is the *Cong* rule: $\dfrac{i{:}k \quad j{:}k \quad i{:}\varphi}{j{:}\varphi}$
- The solution is to impose a "direction": Only one nominal in the equivalence class is saturated. But equality should still be an equivalence, so an irrestricted version for nominals is introduced.

$$\text{(wCong)}\ \frac{i{:}k \quad j{:}k \quad i{:}\varphi}{j{:}\varphi} \qquad \text{(Nom)}\ \frac{i{:}k \quad j{:}k \quad i{:}n}{j{:}n}$$

j is the earliest introduced nominal making k true

Bolander, T. and Blackburn, P.
Termination for Hybrid Tableaus
Journal of Logic and Computation, 17, 517-554, 2007.

A Tableau Based Prover for Hybrid Logics

- A sound, complete and terminating calculus has been implemented in the prover HTab
- Available at `https://hackage.haskell.org/package/HTab`.
- Implemented in haskell, with a GPL license. I.e., you can get the code and change it!
- Demo

What we want to cover

- Introduce the basic hybrid logics $\mathcal{H}(@)$.
- Extend the standard translation to $\mathcal{H}(@)$.
- Talk about expressive power. Bisimulation.
- Define a tableax algorithm for satisfiability of formulas in the basic modal logic (BML).
  - How do we check whether a formula has a model?
  - What can we learn from tableaux?
- Transform the tableau for BML into one for $\mathcal{H}(@)$.
- Discuss more expressive hybrid logics.

Dynamic Naming

- Can we say "the current point is reflexive" in $\mathcal{H}(@)$?

Dynamic Naming

- Can we say "the current point is reflexive" in $\mathcal{H}(@)$? No, if the current point is not named. Nominals and @ gives us control on named states. Unnamed states are beyond $\mathcal{H}(@)$.

Dynamic Naming

- Can we say "the current point is reflexive" in $\mathcal{H}(@)$? No, if the current point is not named. Nominals and @ gives us control on named states. Unnamed states are beyond $\mathcal{H}(@)$.
- If nominals are "like constants," then it would be natural to add "variables" and a means to assign values to them.

Dynamic Naming

- Can we say "the current point is reflexive" in $\mathcal{H}(@)$? No, if the current point is not named. Nominals and @ gives us control on named states. Unnamed states are beyond $\mathcal{H}(@)$.
- If nominals are "like constants," then it would be natural to add "variables" and a means to assign values to them.
- Define formulas of $\mathcal{H}(@, \downarrow)$ as follows.

$$\mathsf{FORM} := p \mid i \mid x \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \Diamond\varphi \mid @_i\varphi \mid \downarrow x.\varphi, \ \ i \in \mathsf{NOM}, x \in \mathsf{VAR}$$

Dynamic Naming

- Can we say "the current point is reflexive" in $\mathcal{H}(@)$? No, if the current point is not named. Nominals and @ gives us control on named states. Unnamed states are beyond $\mathcal{H}(@)$.
- If nominals are "like constants," then it would be natural to add "variables" and a means to assign values to them.
- Define formulas of $\mathcal{H}(@, \downarrow)$ as follows.

$$\mathsf{FORM} := p \mid i \mid x \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \Diamond\varphi \mid @_i\varphi \mid \downarrow x.\varphi, \quad i \in \mathsf{NOM}, x \in \mathsf{VAR}$$

We use hybrid models $\langle W, R, V\rangle, g$, but $g : \mathsf{NOM} \cup \mathsf{VAR} \to W$.

$$\mathcal{M}, g, w \models x \quad \text{iff} \quad w = g(x) \text{ for } x \in \mathsf{VAR}$$

Dynamic Naming

- Can we say "the current point is reflexive" in $\mathcal{H}(@)$? No, if the current point is not named. Nominals and @ gives us control on named states. Unnamed states are beyond $\mathcal{H}(@)$.
- If nominals are "like constants," then it would be natural to add "variables" and a means to assign values to them.
- Define formulas of $\mathcal{H}(@, \downarrow)$ as follows.

$$\mathsf{FORM} := p \mid i \mid x \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \Diamond\varphi \mid @_i\varphi \mid {\downarrow}x.\varphi, \quad i \in \mathsf{NOM}, x \in \mathsf{VAR}$$

We use hybrid models $\langle W, R, V \rangle, g$, but $g : \mathsf{NOM} \cup \mathsf{VAR} \to W$.

$$\begin{array}{rcl} \mathcal{M}, g, w \models x & \text{iff} & w = g(x) \text{ for } x \in \mathsf{VAR} \\ \mathcal{M}, g, w \models {\downarrow}x.\varphi & \text{iff} & \mathcal{M}, g^x_w, w \models \varphi \\ & & g^x_w \text{ ideantical to } g \text{ except that } g(x) = w. \end{array}$$

What can we do with @ and $\downarrow$

A whole of a lot:

- $\downarrow x.\Diamond x$:

What can we do with @ and ↓

A whole of a lot:

- $\downarrow x.\Diamond x$: The current point is reflexive.

What can we do with @ and $\downarrow$

A whole of a lot:

- $\downarrow x.\Diamond x$: The current point is reflexive.
- $\downarrow x.\Diamond(\neg x \wedge \Diamond x)$:

What can we do with @ and ↓

A whole of a lot:

- $\downarrow x.\Diamond x$: The current point is reflexive.
- $\downarrow x.\Diamond(\neg x \wedge \Diamond x)$: The current point is related to anotherm different point which can reach back.

What can we do with @ and $\downarrow$

A whole of a lot:

- $\downarrow x.\Diamond x$: The current point is reflexive.
- $\downarrow x.\Diamond(\neg x \wedge \Diamond x)$: The current point is related to anotherm different point which can reach back.
- $\downarrow x.(\Diamond(\varphi \wedge \downarrow y.@_x\Box(\Diamond y \rightarrow \psi)))$:

What can we do with @ and ↓

A whole of a lot:

- $\downarrow x.\Diamond x$: The current point is reflexive.
- $\downarrow x.\Diamond(\neg x \wedge \Diamond x)$: The current point is related to anotherm different point which can reach back.
- $\downarrow x.(\Diamond(\varphi \wedge \downarrow y.@_x\Box(\Diamond y \rightarrow \psi)))$: Until φ is the case, ψ is the case.

What can we do with @ and $\downarrow$

A whole of a lot:

- $\downarrow x.\Diamond x$: The current point is reflexive.
- $\downarrow x.\Diamond(\neg x \wedge \Diamond x)$: The current point is related to anotherm different point which can reach back.
- $\downarrow x.(\Diamond(\varphi \wedge \downarrow y.@_x\Box(\Diamond y \to \psi)))$: Until φ is the case, ψ is the case.

Good News:

- $\mathcal{H}(@, \downarrow)$ is very expressive.

What can we do with @ and $\downarrow$

A whole of a lot:

- $\downarrow x.\Diamond x$: The current point is reflexive.
- $\downarrow x.\Diamond(\neg x \wedge \Diamond x)$: The current point is related to anotherm different point which can reach back.
- $\downarrow x.(\Diamond(\varphi \wedge \downarrow y.@_x\Box(\Diamond y \rightarrow \psi)))$: Until φ is the case, ψ is the case.

Good News:

- $\mathcal{H}(@, \downarrow)$ is very expressive.
- $\mathcal{H}(@, \downarrow)$ is local (invariant under generated submodels).

What can we do with @ and $\downarrow$

A whole of a lot:

- $\downarrow x.\Diamond x$: The current point is reflexive.
- $\downarrow x.\Diamond(\neg x \land \Diamond x)$: The current point is related to anotherm different point which can reach back.
- $\downarrow x.(\Diamond(\varphi \land \downarrow y.@_x\Box(\Diamond y \to \psi)))$: Until φ is the case, ψ is the case.

Good News:

- $\mathcal{H}(@, \downarrow)$ is very expressive.
- $\mathcal{H}(@, \downarrow)$ is local (invariant under generated submodels).
- Model and Proof theoretically well behaved: sound and complete Hilbert style axiomatization, interpolation, Beth definability, etc.

What can we do with @ and $\downarrow$

A whole of a lot:

- $\downarrow x.\Diamond x$: The current point is reflexive.
- $\downarrow x.\Diamond(\neg x \wedge \Diamond x)$: The current point is related to anotherm different point which can reach back.
- $\downarrow x.(\Diamond(\varphi \wedge \downarrow y.@_x\Box(\Diamond y \rightarrow \psi)))$: Until φ is the case, ψ is the case.

Good News:

- $\mathcal{H}(@, \downarrow)$ is very expressive.
- $\mathcal{H}(@, \downarrow)$ is local (invariant under generated submodels).
- Model and Proof theoretically well behaved: sound and complete Hilbert style axiomatization, interpolation, Beth definability, etc.

Bad News:

- Satisfiability for $\mathcal{H}(\downarrow)$ (i.e., without nominals and @) is already undecidable.